Patent Office
Canberra

CERTIFIED COPY OF
PRIORITY DOCUMENT

I, ANNA MAIJA MADL, ACTING TEAM LEADER EXAMINATION SUPPORT & SALES hereby certify that annexed is a true copy of the Provisional specification in connection with Application No. PQ 1312 for a patent by SILVERBROOK RESEARCH PTY LTD filed on 30 June 1999.

WITNESS my hand this
Fourth day of April 2000

A.M. Madl

ANNA MAIJA MADL
ACTING TEAM LEADER
EXAMINATION SUPPORT & SALES

COMMONWEALTH OF AUSTRALIA
AUSTRALIA
PATENT OFFICE

AUSTRALIA

Patents Act 1990

PROVISIONAL SPECIFICATION

Applicant(s):

SILVERBROOK RESEARCH PTY LTD

Invention Title:

A METHOD AND APPARATUS (NPAGE03)

The invention is described in the following statement:

A METHOD AND APPARATUS (NPAGE03)

Field of the Invention

The present invention relates to the field of transaction authorization.

Background of the Invention

The background of the invention is detailed in the attached appendix A which also sets out a detailed description of the invention.

Summary of the Invention

It is an object of the present invention to provide an improved transaction authorisation system.

In accordance with a first aspect of the present invention, there is provided a method of authorising transactions comprising the steps of: printing a summary of the transaction on a print media; printing an encoded representation of the transaction in an authorisation area on the print media; providing a recipient of the print media with a handheld pen shaped sensor device, for marking their signature in the authorisation area; sensing a series of penstrokes whilst the recipient marks their signature in the authorisation area; sensing the encoded representation of the transaction whilst the recipient marks their signature in the authorisation area; determining, from the sensed penstrokes and the sensed encoded representation, whether to authorise the transaction.

Description Preferred in Other Embodiments

The preferred embodiment of the present invention is an adaption of the technology as set out in the attached appendix A which provides for a detailed description of the implementation of an information distribution and printing system denoted NETPAGE.

Whilst the preferred embodiment can be readily be implemented utilizing all the aspects of the NETPAGE technology, the preferred embodiment is particularly directed to a subset thereof. In particular, to the utilization of the NETPAGE pen for the authorisation of transactions.

As illustrated in Fig. 1, the principles of the NETPAGE pen operation are utilized in the printing out of accounts etc. 1. The printing includes an area 2 for a user to sign should they wish their account to be automatically paid by their credit card. The area 2 includes a NETPAGE encoding of information which uniquely identifies the account. By using a NETPAGE pen to sign the space 2, the pen records the signature key strokes and also the corresponding account information. This information is then forwarded to the issuer of the account as an authorization of payment.

In this arrangement the information printed in area 2 can be printed by any variable printing technology and does not require the utilization of a MEMJET printer. Hence, this provides an immediate use of the NETPAGE pen technology as set out in Appendix A.

It would be appreciated by a person skilled in the art that numerous variations and/or modifications may be made to the present invention as shown in the specific embodiments in the attached documentation without departing from the spirit or scope of the invention as broadly described in the attached appendices. The present embodiments are, therefore, to be considered in all respects to be

illustrative and not restrictive.

We Claim:

1. A method of authorising transactions comprising the steps of:

printing a summary of the transaction on a print media;

printing an encoded representation of the transaction in an authorisation area on said print media;

providing a recipient of said print media with a handheld pen shaped sensor device, for marking their signature in said authorisation area;

sensing a series of penstrokes whilst said recipient marks their signature in said authorisation area;

sensing said encoded representation of said transaction whilst said recipient marks their signature in said authorisation area;

determining, from said sensed penstrokes and said sensed encoded representation, whether to authorise said transaction.

2. A method of authorising transactions substantially as hereinbefore disclosed with reference to the accompanying drawing.

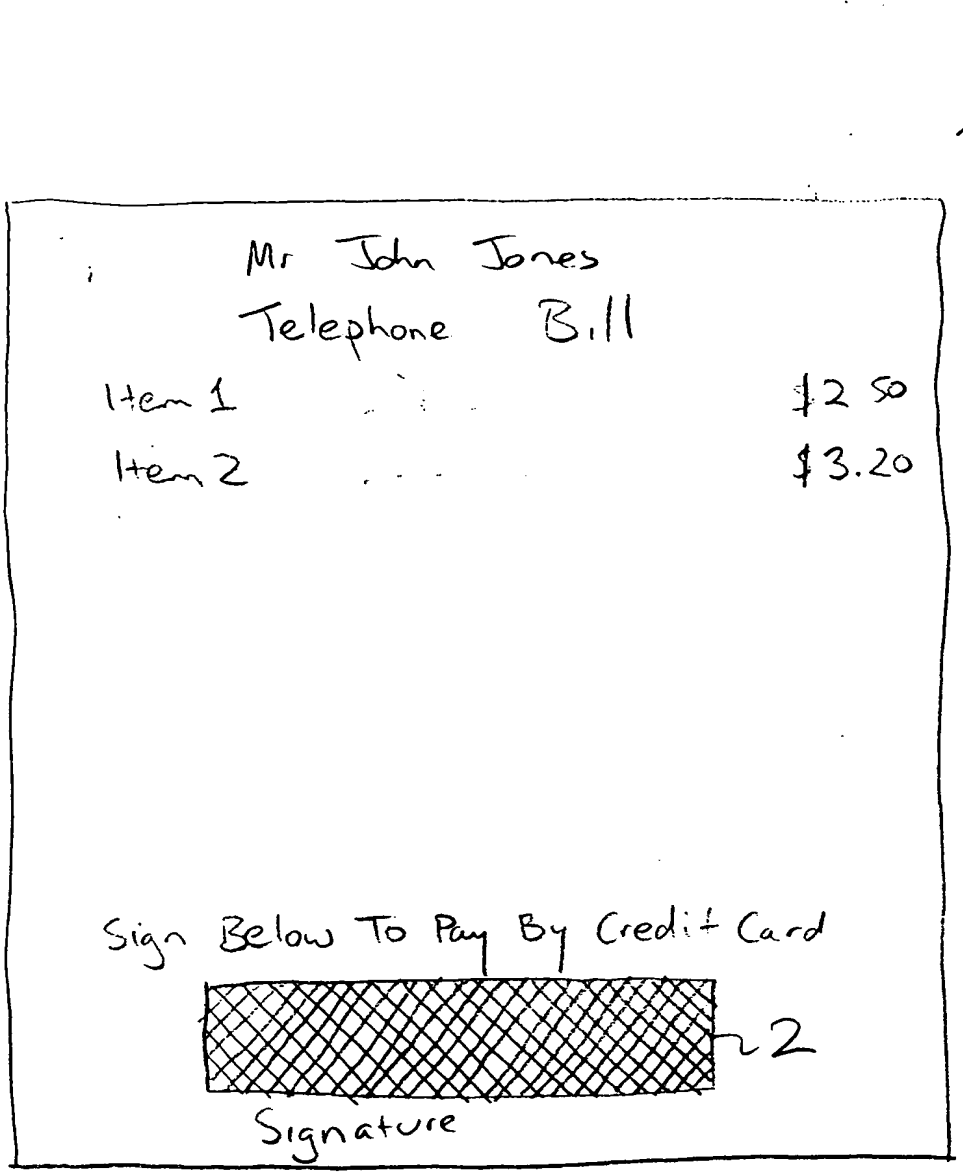

1
Mr John Jones
Telephone Bill
Item 1 ... $2 50
Item 2 ... $3.20
Sign Below To Pay By Credit Card
2
Signature

Fig. 1

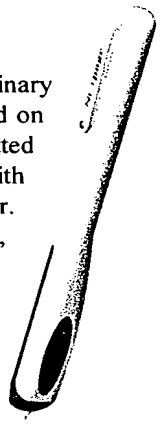
ABSTRACT

A method of authorising transactions comprising the steps of: printing a summary of the transaction on a print media; printing an encoded representation of the transaction in an authorisation area on the print media; providing a recipient of the print media with a handheld pen shaped sensor device, for marking their signature in the authorisation area; sensing a series of penstrokes whilst the recipient marks their signature in the authorisation area; sensing the encoded representation of the transaction whilst the recipient marks their signature in the authorisation area; determining, from the sensed penstrokes and the sensed encoded representation, whether to authorise the transaction.

Appendix A

OVERVIEW

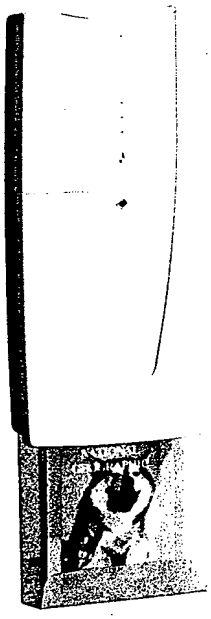
1 Introduction



Netpages are pages of high-quality text, graphics and images printed on ordinary paper, but which work almost like interactive Web pages. Information encoded on each page in invisible ink is picked up by an optically-imaging pen and transmitted to the network. Active "links" and "buttons" on each page can be "pressed" with the pen to request information from the network or signal preferences to a server. Text written by hand on a Netpage is automatically recognized via the pen, allowing forms to be filled in. Signatures recorded on a Netpage are automatically verified, allowing e-commerce transactions to be securely authorized.

The pen, shown on the right, works in conjunction with a Netpage Printer, an Internet-connected printing appliance for home, office or mobile use. The pen is wireless and communicates with the Netpage Printer using an encrypted radio frequency signal.

The Netpage Printer delivers, periodically or on demand, personalized newspapers, magazines, catalogs, brochures and other publications, all printed at high quality as interactive Netpages. Unlike a personal computer, the Netpage Printer, shown on the right, is an appliance typically wall-mounted in the kitchen or near the breakfast table, i.e. the place where the morning news is first consumed, and the household's point of departure for the day. It also comes in tabletop, desktop, portable and miniature versions.



Netpages printed at their point of consumption combine the ease-of-use of paper with the timeliness and interactivity of an interactive medium.

Netpages are crucially enabled by Memjet printing technology [4], which makes high-speed magazine-quality printing affordable to consumers. A Netpage publication has the physical characteristics of a traditional news magazine, i.e. a set of letter-size glossy pages printed in full color on both sides, bound together for easy navigation and comfortable handling. The Netpage Printer prints 60 to 90 full-color Netpages per minute.

The Netpage Printer exploits the growing availability of broadband Internet access. Cable service is available to 95% of households in the United States [75], and cable modem service offering broadband Internet access is already available to 20% of these [13]. The Netpage Printer also operates with slower connections, but with longer delivery times and lower image quality.

Netpage Publication Servers on the Internet deliver print-quality publications to Netpage Printers. Periodical publications are delivered automatically to subscribing Netpage Printers via pointcasting and multicasting Internet protocols. Personalized publications are filtered and formatted according to individual user profiles.

A Netpage Printer supports any number of pens, and a pen can work with any number of Netpage Printers. Each Netpage Pen has a unique identifier. A household may have a collection of colored Netpage Pens, one assigned to each member of the family. This allows each user to maintain a distinct profile with respect to a Netpage Publication Server. A Netpage Pen can also be registered with a Netpage Registration Server and linked to one or more payment card accounts. This allows e-commerce payments to be securely authorized using the Netpage Pen. The Netpage Registration Server compares the signature cap-

tured by the Netpage Pen with a previously registered signature, allowing it to authenticate the user's identity to the e-commerce system. Other biometrics can also be used to verify identity. A version of the Netpage Pen includes fingerprint scanning, verified in a similar way by the Netpage Registration Server.

Although a Netpage Printer delivers periodicals such as the morning newspaper without user intervention, it never delivers unsolicited junk mail. It only delivers periodicals from subscribed or otherwise authorized sources. The Netpage Printer is unlike a fax machine or e-mail account which is visible to any junk mailer who knows the telephone number or email address.

1.1 PUBLISHING TO NETPAGE PRINTERS

There are a number of advantages to publishing to Netpage Printers. The magazine quality of Netpage Printer output makes it a more attractive publishing and advertising medium than both traditional newsprint and computer screens.

The cost of paper and ink consumption is transferred to the user. Subscription fees can be eliminated entirely in lieu of the user taking on this extra cost, and the user thus perceives a better-value product. The erratic price of newsprint is removed from publishers' profit equations, resulting in more stable margins. A new market for paper and ink consumables, with its own margins, is created.

The cost of consumables can be selectively subsidized, for example when non-editorial publications such as product brochures and account statements are printed.

Capital and maintenance expenditure on printing plant is effectively transferred to the user, although the perceived expense is small because Netpage Printers are sold at close to cost or given away to encourage adoption, subsidized by future advertising profits. Maintenance can also be subsidized or its cost included in a longer term service agreement.

Costly physical distribution is replaced by electronic distribution via a preexisting and widely subscribed network - the Internet.

Both the editorial and advertising content of publications delivered via the Netpage Network can be customized for each user. Editorial content can be personalized according to the user's profile. Advertising can be localized to the user's locality and optionally targeted to the user's demographic.

A personalized publication can be a small fraction of the size of its traditionally-delivered counterpart, yet contain the same amount of information relevant to the user, and in a more accessible form. The user appreciates the more efficient and digestible publication.

Localized advertising can be targeted to more specific localities and their associated demographics, and this allows advertising space to be exploited more efficiently, i.e. with less waste. Advertisers are constantly pressing traditional publishers for greater localization, something which they have great difficulty delivering cost-effectively.

By revealing personal information such as age, gender, marital status, income, profession, education, etc., the user can allow the advertising to be more carefully targeted. In return they can receive greater subsidies and discounted product offers. As advertising becomes more targeted, it becomes less of a nuisance and more of a service in itself.

Although the publication's per-page circulation figures fall drastically, the publication's actual per-section readership is preserved, and the correspondingly higher advertising rates for personalized delivery can exactly compensate for this.

Advertising delivered via the Netpage Network has the dual benefits of print and online delivery. Print supports the impact of large-format ads. Online delivery supports customization, linking, and measurability, and consequently online charging models.

Consider a full-page advertisement for a new car model in a news magazine delivered via the Netpage Network. The advertising campaign can be national or even international. The ad only appears if compatible with the user's demographic, either implied by their ZIP code or more explicitly by their personal details. Anyone who requests a product brochure via the on-ad button receives one immediately via their Netpage Printer, customized with a list of local dealers. If they press a particular dealer's "contact me" button in the brochure, the dealer receives a message via the system and contacts the user by telephone.

The publisher profits in the normal way by selling the advertising space, but can also profit by receiving a fee on the "click-through" to the brochure, and a commission on any product sale which eventuates.

The Netpage Network promises to be the most effective advertising medium ever conceived. It combines the editorial and print quality of traditional publications with arbitrarily finely targeted advertising, and provides a direct link between advertising, product information, and purchasing. Added revenue from click-through fees and e-commerce commissions may even allow users' costs - printer, ink, paper, and Internet access - to be fully subsidized.

2 The Demise of Paper

Online publication has many advantages over traditional paper-based publication. From the consumer's point of view, information is available on demand, information can be navigated via hypertext links, information can be searched, and information can be automatically personalized.

From the publisher's point of view, the costs of printing and physical distribution are eliminated, and the publication becomes more attractive to the advertisers who pay for it because it can be targeted to specific demographics and linked to product sites.

Online publication also has a few disadvantages. Computer screens are inferior to paper. At the same quality as a magazine page, an SVGA computer screen displays only about a fifth as much information[1]. Both CRTs and LCDs have brightness and contrast problems, particularly when ambient light is strong. Ink on paper, being reflective rather than emissive, is both bright and sharp in ambient light.

Faced with reading more than the most trivial amounts of text on a screen, most people prefer to print it before reading it. Increasingly, online publishers are recognizing this and providing information in formats suitable for printing. At one extreme this means providing text-only versions of documents so they print efficiently, i.e. without imposing a screen format on the printed page; at the other extreme it means providing formatted versions of documents - e.g. in Adobe's Portable Document Format (PDF) - so they print at high quality.

Editorial content is often compromised to fit the online medium and the habits of the people who frequent it, who tend to browse rather than read. Although powerful new advertising models become possible, it becomes more difficult to deliver effective advertisements.

To truly enable online publication, many people envisage a universal information appliance - a lightweight portable "tablet" with a page-size touch-sensitive color display and a high-bandwidth wireless connection to the Internet. First proposed by Xerox PARC's Alan Kay in the mid 1970s in the form of the "Dynabook", and partially realized in recent paperback-sized electronic books, even Bill Gates is now confidently predicting that such a device will soon augur the death of paper publications [30].

To achieve low power consumption, low weight, and paper-like display quality, a bistable reflective display technology is required. Several candidates are now emerging from the labs, including Kent Display's cholesteric LCD technology (chLCD) [48], Xerox' "Gyricon" rotating ball technology [35], and E Ink's electrophoretic technology [26,62]. ChLCD is arguably closest to practical deployment [25].

Next-generation cellular phone networks promise 2Mbps packet switching [27], comparable to the broadband access people are getting used to in cable networks [59]. Satellite networks, while offering or promising still higher speeds [39,65,81], require receivers difficult to deploy in mobile devices.

Beyond the vision of the basic tablet, E Ink imagines its digital ink "printed" onto a number of flexible pages bound into a book, preserving the physical navigability of a paper-based publication, and approaching its low cost, but allowing the pages to be rewrit-

1. Assuming a magazine page has an equivalent digital resolution of 200 continuous-tone pixels per inch.

ten electronically in place. They optimistically predict newspapers delivered in this way within five years [21], despite fundamental problems yet to be overcome [25].

The advantages of a tablet are many. Unlike a desktop or notebook computer, a tablet may actually provide a pleasant reading experience. Unlike a paper publication, a tablet provides intelligent access to an unlimited amount of information; its weight is not dictated by the amount of information it carries. More than just an information appliance, it can also act as a multi-purpose multimedia communications device and interactive entertainment device.

A tablet has disadvantages too. It uses batteries which run down and have to be recharged. It may break when dropped or malfunction when exposed to hot coffee. It's not quite cheap enough to be disposable - so there's still a problem if it's misplaced or stolen. It has a "user interface" which has to be learned. The leading candidate display technology - chLCD - is still less than half as reflective (i.e. "bright") as paper.

The drawbacks of traditional paper-based publications have little to do with paper itself, and much to do with how the information gets onto the printed page. The economics of centralized printing and distribution prevent the kind of information selection, personalization and navigation people have come to expect from interactive electronic media such as the Internet. The inefficiency of printing and distributing a hundred-page newspaper to a customer who may read only a few pages is widely decried.

Given a technology such as Memjet, it becomes economic to print high-quality publications at their point of consumption rather than at their point of production. The Netpage Printer leverages Memjet to deliver personalized publications to the home, gaining many of the advantages of online publication, while retaining the ease-of-use of high-quality printed ink on paper.

Netpages and the Netpage Printer address the key problems of online publication, without relying on the development and consumer acceptance of a new reading device.

3 News and Advertising Trends

3.1 NEWSPAPERS

People obtain news from a variety of sources - network and cable television, radio, daily newspapers, and weekly newsmagazines. In the United States, although the various news media are healthy and profitable, per capita news consumption is somewhat in decline as a new generation of young adults have less time to read and favor television entertainment over news [22]. Yet six out of ten adult Americans read a newspaper every day [68].

The United States has about 1500 daily newspapers with a total circulation of 57 million. Just the top ten "national" dailies (Wall Street Journal, USA Today, New York Times, Los Angeles Times, Washington Post, etc.) account for a circulation of 10 million. The major weekly newsmagazines (Time, Newsweek, U.S. News) have a similar (weekly) circulation of about 10 million.

In 1997, newspaper companies' revenue exceeded $24 billion, a five-year high, and margins nudged 20% [51], due both to increased spending on advertising and to reduced prices of newsprint.

Television and radio, by their nature, excel at delivering breaking news. Newspapers and newsmagazines, on the other hand, deliver the depth and analysis behind the headlines. Broadcast news in isolation does a poor job of informing the public. The more local the news is, the poorer the broadcast coverage, and the greater the public's dependence on newspapers.

Newspaper content and packaging has evolved considerably since the 1970s. News is somewhat softer, news stories are shorter and more well-written, there are more feature articles, and there is more editorial and reader opinion. Newspapers are more structured. Identifiable sections make them more accessible, and provide greater focus for advertisers. Much special-interest content has migrated from daily inclusion to weekly sections. These cover topics such as lifestyle, personal finance, entertainment, technology, etc. The proportion of graphics and pictures is greater. Color is widely used. Newspapers are easier to use and more entertaining than ever before, if at the expense of some "hard" news.

Daily newspapers are growing increasingly dependent on the various wire services. A newspaper may excel at local and regional news, but rely on the major wires (Associated Press and United Press International) for national and international news, the so-called "supplemental" wires (LA Times/Washington Post, NY Times, Scripps-Howard, Knight-Ridder Tribune, etc.) for specific strengths (and value-for-money), and the international wires (Reuters etc.) for international perspective. A growing number of newspapers operate more as news aggregators than news gatherers.

Advertising typically contributes more than 75% of newspaper and magazine revenue, while subscriptions contribute less than 25% [34,66,67]. National advertising makes up roughly 14% of advertising spending, retail advertising 46%, and classified advertising 40% [67].

Advertisers are pursuing increasingly specific targeting, favoring quality newspaper readership over raw circulation [29], and using more targeted media where possible. Magazines, for example, have more specific readerships than newspapers, free "shoppers" are

very localized by their nature, while direct mailers can target demographics based on individually-categorized ZIP codes, or databases of individuals.

Newspapers have responded with geographically zoned editions to support local advertising, and greater sectioning of their product. They have also expanded their page counts to provide more advertising scope, despite erratic newsprint prices in the 1990s [8].

Despite this, there is ongoing conflict between newspapers' mass distribution model, and advertisers' need for micro-targeting [33]. This conflict, coupled with advertisers' desire for higher-quality printing of color images, is motivating a shift from run-of-press (ROP) advertising to inserts [67]. The downside to inserts is that editorial context is lost.

3.2 Online News Delivery

Fearing the online migration of advertising, traditional news publishers from both broadcast and print have ventured into Internet-based news delivery, wanting to establish a presence at whatever cost before newcomers become entrenched. Most newspapers are still reporting losses from their online operations [67].

Online news delivery offers a number of advantages. Breaking news can be delivered as soon as it happens. News can be customized for individual readers according to their preferences and geographic locations. Readers can explore stories to arbitrary depth, follow links to related resources, and search archival material. Readers can participate in discussion groups and contribute to opinion polls. The news itself can incorporate audio and video clips, and can include live transmissions, converging with broadcast.

Online news delivery also has disadvantages. Computer screens are of limited size and quality compared with print. Few people enjoy reading a story of any length on a computer screen. Computers are not portable in the wide sense that a newspaper is. The news may be more timely, but the time and place in which it can be consumed are more constrained than with a newspaper.

Despite the power of hypertext, many online readers express a preference for a linear presentation, "where they [can] skim one section after another until the presentation [is] exhausted" [15]. Interestingly, a majority of traditional newspaper readers admit they scan every page in the main section of the newspaper [68], looking for items of interest without necessarily knowing what they're looking for, and achieving some kind of closure at the end. Online hypertext, by contrast, is both a limitless resource and a bottomless pit.

While traditional news publishers such as The New York Times can deploy full editorial content online [82], newcomers such as Yahoo typically only provide "raw" news items sourced from the wire services [63].

A recent survey indicates that 21% of the 74 million Internet users in the United States regularly read news online as an alternative to traditional print and broadcast sources, and 16% obtain a major proportion of their news online [69]. More broadly, between 37% and 64% of the Internet population reads news online at least once a week. The fluctuations in the figures are related to what may be happening in the news. Major or breaking news stories attract more users - 46% of Americans say they only follow national news stories when "something major is happening" [28].

With 41% of Americans online, the Internet population has become mainstream, and the weather has become the most popular news online. This is closely followed by technology

news, entertainment news, and local news. As one observer puts it, all of this "sound[s] like the 6 o'clock news" [69]. As a reflection of these habits, the online audience share of national newspapers has diminished from 23% in 1995 to 16% in 1998, while the online audience share of broadcast TV sites has grown.

3.3 ONLINE ADVERTISING

At its simplest, advertising alerts a motivated customer to the availability of a product, possibly at a competitive price. At a more sophisticated level, advertising seeks to influence future purchasing decisions by creating brand awareness. Ultimately, advertising seeks to create desire for a product even when actual need is absent.

Advertising prices are traditionally based on how many people see the advertisement, and their spending power in relation to the product. In practice, the more homogeneous the demographics of the audience, the easier it is to match to a product, and hence the higher the corresponding advertising cost per thousand (CPM). Broadcast media use ratings and timeslot demographics to set advertising rates. Print media use audited circulation figures and sectional readership demographics.

The simplest online advertising model is also based on how many people see the ad. Online this has the advantage of being based on solid numbers, since the number of "impressions" of a particular Web page can be counted exactly.

The specific advantage of an online ad, however, is that the ad itself can measurably capture a sales lead by acting as a link to a product site. The product site may simply provide more product information in the form of specifications, pricing, and ordering details. It may also support immediate online ordering, thus completing the link from ad to sale. Beyond providing simple ad exposure, it is this measurable linking of advertisement to sales lead or sale which is the strength of online advertising [73]. Cost per click (CLC) charging is gaining acceptance but is still controversial.

Beyond CLC, there exists the possibility of paying a commission to the ad host on any sale that actually eventuates [76]. Amazon.com is probably the best-known example of a company paying commissions to other sites in this way.

The broader advantage of online advertising is that advertising can be localized and targeted arbitrarily finely, in conjunction with the publication of online content such as news. This is the strategy pursued by online advertising agencies such as Click~Through [19], which acts as the middle-man between advertisers and online content publishers. They expect online advertising to represent more than 10 percent of all advertising revenue by 2001.

Since online ads are necessarily small-format, they communicate best with motivated customers already on the look-out for a particular product or service. Online ads are less suited to building brand awareness or creating buying desires, since the real substance of the advertisement - the product Web site - is a click away from the initial small-format ad. A small-format online ad can't provide the single-hit emotional impact of a large-format print ad, and conversely, the online world can't support the large-format ads that print can.

So-called interstitial ads, which appear full-screen when traversing from one page of information to the next, go some way to providing a medium for larger-format ads online [76]. User resistance, however, seems to be preventing their widespread use.

3.4 ONLINE CLASSIFIED ADVERTISING

Classified advertising is indisputably suited to online delivery. Unlike their traditional printed counterparts, online classifieds can be easily searched, and are not subject to space constraints. The online migration of classified advertising is considered a serious threat to newspapers' classified advertising profits [78], and some newspapers are building an online presence for this reason alone. Some observers predict as much as 50% of classified advertising revenue moving online within the next ten years [98].

Another problem faced by newspapers, who rely on classifieds for up to 40% of advertising spending, is that many newcomers are offering free online classified advertising as a way of building a venue for non-classified advertising.

4 News Personalization

From the reader's point of view, a personalized news publication can provide more information in fewer pages. The actual form this personalization takes, however, is not necessarily obvious.

The MIT Media Lab's News in the Future (NiF) project has been championing the concept of "The Daily Me" for almost two decades [60]. Nicholas Negroponte, one of the project's founders, envisages a highly personalized news publication which is no longer driven by "what other people think is news" [64]. By way of examples close to his own needs, it includes news about people and places about to be encountered, and puts "the most important [news] of all" - a summary of e-mail - on the front page [11]. Negroponte recognizes the need to vary the degree of personalization, advocating a higher "serendipity factor" on a lazy Sunday than on a working weekday.

The opposing view holds that the value of a news publication lies precisely in its *shared* nature. It reflects the common concerns and values of a community of readers, and establishes a baseline of expectations of what they are all supposed to know [89]. As a consequence, the publication also speaks with a consistent editorial voice and with consistent assumptions about the reader's level of background knowledge. Such a shared publication allows its readers to orient themselves in relation to their community.

NiF's Walter Bender answers the charge (in his own words) of "The Daily Me engendering a fragmented world populated by self-interested myopes", by stressing the possibilities of personalizing individual news items [9]. This can consist in varying the depth of an item, or supplementing it with background information, based on the reader's level of knowledge. It can involve interpreting information relative to the reader's background, such as (somewhat dubiously) making value judgements about the weather relative to the reader's normal home town weather. It can also be as simple as using metric rather than imperial units.

FishWrap [15,61], MIT's personalized campus newspaper and NiF's latest offering, goes further by creating a front page whose content represents an explicit community consensus. Each front page news item is prioritized according to the number of readers who put it forward for inclusion. The rest of the newspaper is still personalized according to each reader's profile, consisting of reader-defined sections containing topics of interest.

There are two implications of recognizing the shared nature of news. Firstly, some news is news to everybody in a community, no matter how personalized they claim they would like their news to be. This implies that the community must make decisions about news item priority, either directly (as in FishWrap) or indirectly via a proxy (i.e. an editor).

Secondly, a news item can only be properly understood in the context of the community for which it is intended. This implies that a news item must be branded with its source (assuming that the source implies the intended target). As an example, it is significant whether a news item regarding the proof of Fermat's last theorem is branded with New Scientist or The New York Times. To a professional mathematician, the latter implies, by its very existence, that the proof is of significance beyond the scientific community.

Of course, a news item must also be branded to allow its source to build and maintain its brand. The brand then allows the reader to infer the quality of the news item from the known quality of the source.

Most personalization of news uses *feature-based filtering*. This means that news item content is matched to topics and keywords in the reader's profile. News sources tag the items they produce with various information to allow them to be effectively filtered. This tagging may be brief or extensive, and may include such things as news item urgency, byline, news category, subject(s), keyword(s), date and time, and location [44]. The body text of a news item can also be scanned directly for keywords, but this may result in false matches if keywords are interpreted out of context. Items in the text such as personal names and locations can be tagged to reduce such ambiguity [44]. Similarly, dates, times, and monetary amounts can be tagged to allow localized presentation.

Feature-based filtering suffers from a number of problems. Filtering based on tags is only as good as the original tagging. The latest tagging standards are only just beginning to be adopted [12]. Filtering based on the text itself is constrained by the intelligence of the text parsing. If based simply on keyword matching, it can be both inaccurate, generating false matches because of word sense ambiguity, and imprecise, generating false mismatches because of a lack of inference.

Feature-based filtering is incapable of discerning more abstract attributes such as quality, style, and point-of-view (unless they're indicated by tags). And since it only matches items anticipated by the user's profile, it is a poor generator of serendipitous finds.

FishWrap's front page comes into existence based on a crude form of *collaborative filtering*. In its broader form, collaborative filtering involves sharing recommendations (or ratings) among *like-minded* people [74]. This means that one person's ratings influence another person if and only if the two share similar interests, i.e. they have similar rating histories. Collaborative filtering overcomes many of the problems of feature-based filtering, since ratings originate with people who have digested the items in question, rather than from automated analysis of the items. Collaborative filtering sidesteps the issue of *explaining* why a person might like a particular item.

Collaborative filtering has problems of its own. The system only works if people are willing to contribute ratings. In contributing ratings, of course, they are both doing the community a service and tuning their own interest profiles. The statistical error in correlating people's interests decreases as the number of ratings increases. However, incentives may have to be offered to encourage people to contribute ratings.

To bootstrap the accumulation of ratings for new items, an independent mechanism must exist to distribute them to a critical mass of people. Conversely, to bootstrap the accumulation of interest profiles for new users, an independent mechanism must exist to distribute a critical mass of items to them.

To allow meaningful accumulation of ratings, a sufficient period of time must be allowed to elapse. This may conflict with the timely delivery of items in question.

The statistical correlation between different people's interests, represented by their rating histories, is most meaningful when the ratings apply to homogeneous items. For a set of heterogeneous items, collaborative filtering is best applied to homogeneous subsets.

In a news setting, collaborative filtering is best applied to feature articles. Features have the longer life span required to support the accumulation of ratings, and are often appreciated for abstract qualities best singled out by collaborative filtering (good writing, humor, incisiveness, etc.).

Naturally, the larger a publication's readership, and the better its taste in relation to a potential reader, the stronger the publication's brand will appear to that reader.

Although it's easy to become preoccupied with automatic filtering, in reality there's more to editing the news than just filtering news feeds. An editor also solicits news, commissions analysis, and offers opinion, ideally ensuring that the publication offers a balanced and complete view of the world.

Perhaps the most important personalization step a reader takes is in selecting a particular publication from a set of available publications, based on its perceived quality and relevance.

Thus the publication's brand equates to the highest-level and most useful filter of all.

ARCHITECTURE

5 Netpage System Architecture

5.1 THE INTERNET

The Internet is a worldwide collection of interconnected networks which communicate using the TCP/IP protocol suite [77]. A TCP/IP-based internetwork not connected to *the* Internet is often referred to as *an* internet (i.e. with a lower-case 'i'). When an internet is deployed within an organization, it is often termed an intranet.

Access to the Internet is widespread in developed countries. In the United States, for example, 41% of the population has access to the Internet [69].

While most consumers still access the Internet via low-speed dial-up modems connected to the switched telephone system, inexpensive broadband access is becoming available to a growing number of households via the cable networks. Cable service is available to 95% of American households [75], and cable modem service is available to a 20% subset [13]. While dial-up modems offer speeds of up to 56Kbps, cable modems offer practical speeds of up to about 3Mbps[1], i.e. over 50 times faster.

DSL (Digital Subscriber Line) [16,17], while offering similar speeds to cable modems but via the telephone system, is not yet widely used. ISDN (Integrated Services Digital Network), although widely used for corporate access, has had little consumer impact due to its high price and comparatively low performance.

The deployment of third-generation (3G) cellular telephony within the next few years will bring practical mobile broadband speeds of 2Mbps [27]. 3G cellular uses WCDMA (wideband code-division multiple access), a spread-spectrum technology. Satellite systems are arguably closer to offering even faster broadband Internet access [39,65,81].

The core of the Internet is made up of a number of independent high-speed fiber-optic networks connected into NAPs (Network Access Points) or peered directly. These have until recently used single-wavelength TDM (Time-Division Multiplexing) SONET (Synchronous Optical Network) transmission systems which utilize about 1% of an optic fiber's capacity to yield a 2.5Gbps OC-48 channel[2]. Carriers are now beginning to deploy multi-wavelength DWDM (Dense Wavelength-Division Multiplexing) systems which yield up to 40 such channels per optic fiber, thus increasing network capacity significantly without requiring the laying of more fiber [14,52]. Internet architects are therefore now contemplating aggregate capacity in the terabit (Tbps) range.

The Internet uses the four-layer TCP/IP protocol suite. The application layer provides various end-to-end application services, and is a client of the transport layer which provides end-to-end delivery services. The transport layer in turn is a client of the network layer which provides packet routing. The network layer is a client of the link layer which encapsulates specifics of the protocols and hardware of the actual communications links.

The core Internet transport protocol, TCP (Transmission Control Protocol), provides a reliable end-to-end delivery service. The core Internet network protocol, IP (Internet Protocol), provides an unreliable and connectionless packet routing service. IP may lose or

1. Although the cable supports 30Mbps and the cable modem theoretically supports 10Mbps.
2. SONETchannels have an OC-*n* designation, where OC stands for Optical Carrier, and *n* gives the channel speed in units of about 52Mbps. An OC-48 channel therefore has a speed of about 2.5Gbps.

deliberately discard packets, and may deliver packets out of order, and it is the responsibility of a higher layer to provide a reliable end-to-end service.

With the proliferation of streaming media services on the Internet, support for multicast is spreading rapidly. Multicast is a form of broadcast with a specific set of recipients. It makes efficient use of network capacity because a packet traverses a network link once rather than once per recipient. It is particularly efficient if the recipients are connected to the Internet via an intrinsically broadcast medium such as cable or satellite. The @Home cable network has successfully enabled multicasting of streaming media services [50].

IP Multicast is an extension of IP, and so is unreliable. While this is often acceptable for time-critical data such as streaming video, it may not be acceptable for other shared data types. Significant effort is being expended to develop reliable multicast transport protocols on top of IP Multicast. Although several reliable multicast protocols are available and have been deployed [55,56,41,42], the Internet standardization process is incomplete [43].

5.2 NETPAGES AND NETPAGE DOCUMENTS

Netpages are the foundation on which a Netpage Network is built. They provide a paper-based user interface to published information and interactive services.

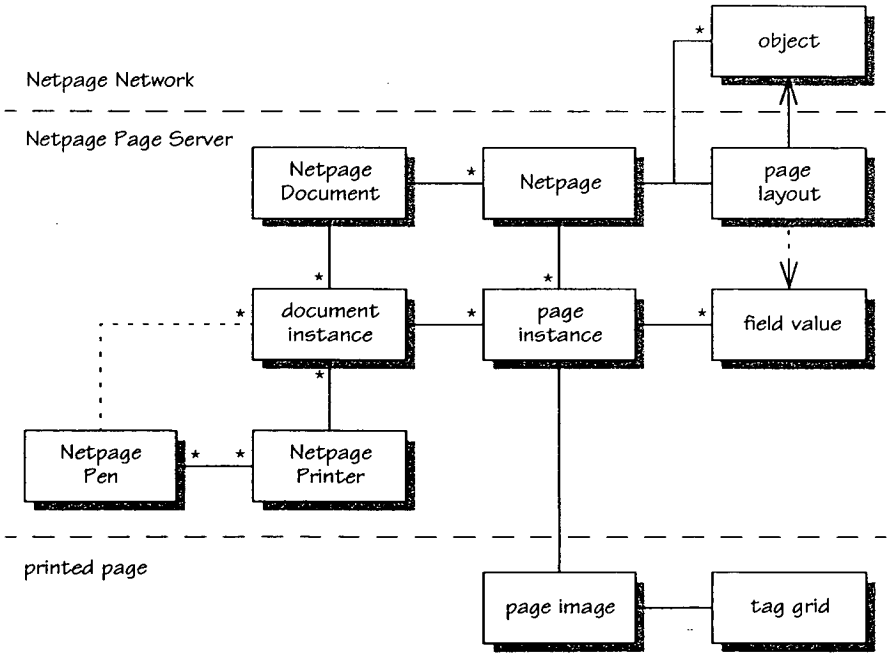


Figure 1. Netpage Document structure (* indicates an n-ary relationship)

Each Netpage consists of a compact page layout maintained persistently by a Netpage Page Server. The page layout refers to objects such as images, fonts and pieces of text, typically stored elsewhere on the Netpage Network.

Netpages are organized into Netpage Documents. Both Netpages and Netpage Documents are assigned globally unique identifiers.

Each Netpage Document has a set of document instances, each of which describes a printed instance of the document. Each Netpage in the Netpage Document has a corresponding set of page instances, each of which describes a printed instance of the page. Both page instances and document instances are assigned globally unique identifiers. They are also uniquely associated with the printer on which they are printed, and the pen which initiated the print request, if known.

Each page instance maintains a set of user-supplied values for fields in the page layout. This ensures that user input is captured and stored independently for each page instance. The separation of page instances and Netpages is crucial for pages which contain input fields, i.e. forms. It is not crucial for pages devoid of input fields, but still useful because it supports independent auditing of each page instance.

The physical page image includes encoded information which identifies the page instance and hence the Netpage to which it corresponds. It also includes encoded information which superimposes an addressable spatial grid over the page image, to allow pen actions performed relative to the page image to be correlated with the contents of the page layout.

The encoded information is normally printed in infrared-absorptive ink on any normal paper substrate which is infrared-reflective. Near-infrared wavelengths are invisible to the human eye but are easily sensed by a solid-state image sensor with an appropriate filter.

The encoded information is picked up by an infrared-imaging pen and transmitted to the associated Netpage Printer. The pen is wireless and communicates with the Netpage Printer using an encrypted radio frequency signal.

The encoded information is organized as a set of tags, each containing both the id of the page instance and the position of the tag. The tags tile the entire page image, and are sufficiently small and densely arranged that the pen can reliably image at least one tag even on a single click on the page. It is important that the pen recognize the page instance id and position on every interaction with the page, since the interaction is stateless.

The tags are error-correctably encoded to make them resilient to errors introduced by dirt on the page or during the imaging process.

Memjet-based Netpage Printers are designed to print a tag grid using infrared (IR) ink. Printers not enabled for IR printing have the option to print tags using IR-absorptive black ink, although this restricts tags to otherwise empty areas of the page. Although such pages have more limited functionality than IR-printed pages, they are still classed as Netpages.

5.3 THE NETPAGE NETWORK

A Netpage Network consists of a distributed set of Netpage Publication Servers, Netpage Page Servers, and Netpage Printers connected via an internet. In technological terms this document describes *any* Netpage Network. In business terms it usually refers to *the* Netpage Network connected via *the* Internet.

As indicated above, a Netpage Page Server maintains persistent information about Netpage Documents, Netpages, and their printed instances, to allow pen operations on printed pages to be interpreted intelligently.

The Netpage Network includes any number of Netpage Page Servers, each handling a subset of Netpages. As indicated above, each page instance is identified by a globally unique

id which is encoded in the tag grid of the corresponding printed page. The Netpage Printer uses this id to retrieve the page layout of the page from a Netpage Page Server when it needs to interpret pen operations relative to the page.

The Netpage Printer uses the internet Distributed Name System (DNS) to resolve a Netpage instance id into a page instance maintained by a particular Netpage Page Server.

The DNS is a protocol and a hierarchical system of name servers used to resolve internet domain names into resources. Planned enhancements to the DNS allow it to be used to resolve more general Uniform Resource Identifiers (URIs), and in particular Uniform Resource Names (URNs), into resource locations [24]. Netpage instance ids are formulated as URNs, allowing the enhanced DNS to be used to resolve them. In the absence of timely standardization and deployment of an enhanced DNS on the Internet, the Netpage Network can deploy its own system of enhanced name servers.

A Netpage Publication Server is an internet server which publishes Netpage Documents to Netpage Printers. It is described in Section 6.

5.4 THE NETPAGE PRINTER

The Netpage Printer is the appliance which prints Netpage Documents. It is connected to a Netpage Network via an internet, ideally via a broadband connection.

Apart from identity and security settings in non-volatile memory, the Netpage Printer contains no persistent storage. As far as a user is concerned, *the network is the computer* [79]. Netpages function interactively across space and time with the help of the distributed Netpage Page Servers, independently of particular Netpage Printers.

The Netpage Printer receives Netpage Documents from Netpage Publication Servers. Each document is distributed in two parts: the page layouts, and the actual text and image objects which populate the pages. Because of personalization, page layouts are typically specific to a particular subscriber and so are pointcast to the subscriber's printer. Text and image objects, on the other hand, are typically shared with other subscribers, and so are multicast to all subscribers' printers.

The Netpage Publication Server optimizes the segmentation of document content into pointcasts and multicasts. After receiving the pointcast of a document's page layouts, the printer knows which multicasts, if any, to listen to.

Once the printer has received the entire document's page descriptions, i.e. page layouts *and* objects, it can print the document.

The printer rasterizes and prints odd and even pages simultaneously on both sides of the sheet. It therefore contains duplexed print engines and imaging units.

The printing process consists of two decoupled stages: rasterization of page descriptions, and expansion and printing of page images. The raster image processor (RIP) consists of one or more standard DSPs running in parallel. The duplexed print engines consist of custom processors which expand, dither and print page images in real time, synchronized with the operation of the printheads in the imaging units.

There are four major design variations embodied in the various Netpage Printer models:

- *form factor*: pocket, portable, desktop, wall-mount or tabletop
- *printhead width*: 4" (photo), 8½" (portrait Letter) or 11" (landscape Letter)
- *paper source*: cut sheet or print cartridge
- *Internet connection*: wired or wireless

This form factor variations yield five basic models, each with variants determined by printhead width (and hence printing speed), and paper source. Eight planned models are defined in Table 1, and illustrated in Figure 2.

Table 1. Netpage Printer models

model	form factor	variant	printhead width	paper source	Internet connect
Microprinter	pocket	R	4"	cartridge	wireless
Travelprinter	portable	R	8½"	cartridge	wireless
Deskprinter	desktop	R		cartridge	wired
Wallprinter	wall-mount	-		cut sheet	or
		Pro	11"	cut sheet	wireless
		Pro R		cartridge	
Tableprinter	tabletop	Pro		cut sheet	
		Pro R		cartridge	

The Deskprinter, Wallprinter and Tableprinter models can be factory-configured with various network modules, allowing both wired and wireless versions. The Microprinter and Travelprinter both use a cellular telephone module, with the promise of broadband speed within a few years.

The Wallprinter models are ideal for unobtrusive installation in a home, while the Tableprinter models might be preferred in an office environment. Note that the Tableprinter models are Wallprinter models factory-adapted for tabletop use via a stand. The Deskprinter, with its small footprint, is ideal for both home and office use.

The Microprinter prints normal Netpages at quarter size, and provides full wireless Netpage Network access in a pocket device.

The paper roll cartridge contains both paper and ink. The paper is in the form of a continuous roll, cut on demand by the printer. The 11" print cartridge has a capacity of 1000 Letter sheets. It also contains the glue supply for binding the sheets of a document together. The 8½" print cartridge has a capacity of 50 Letter sheets, or equivalently 100 A5 sheets. The 4" print cartridge has a capacity of 36 6×4 photos, or 41 quarter-size Netpages. The 8½" and 4" print cartridges don't contain a glue supply because neither the Microprinter nor the Travelprinter includes a binding mechanism.

The 4" printhead models print at 30 quarter-size pages per minute. The 8½" printhead models print at 60 pages per minute, or 30 duplex sheets per minute. The 11" printhead models print at 90 pages per minute, or 45 duplex sheets per minute.

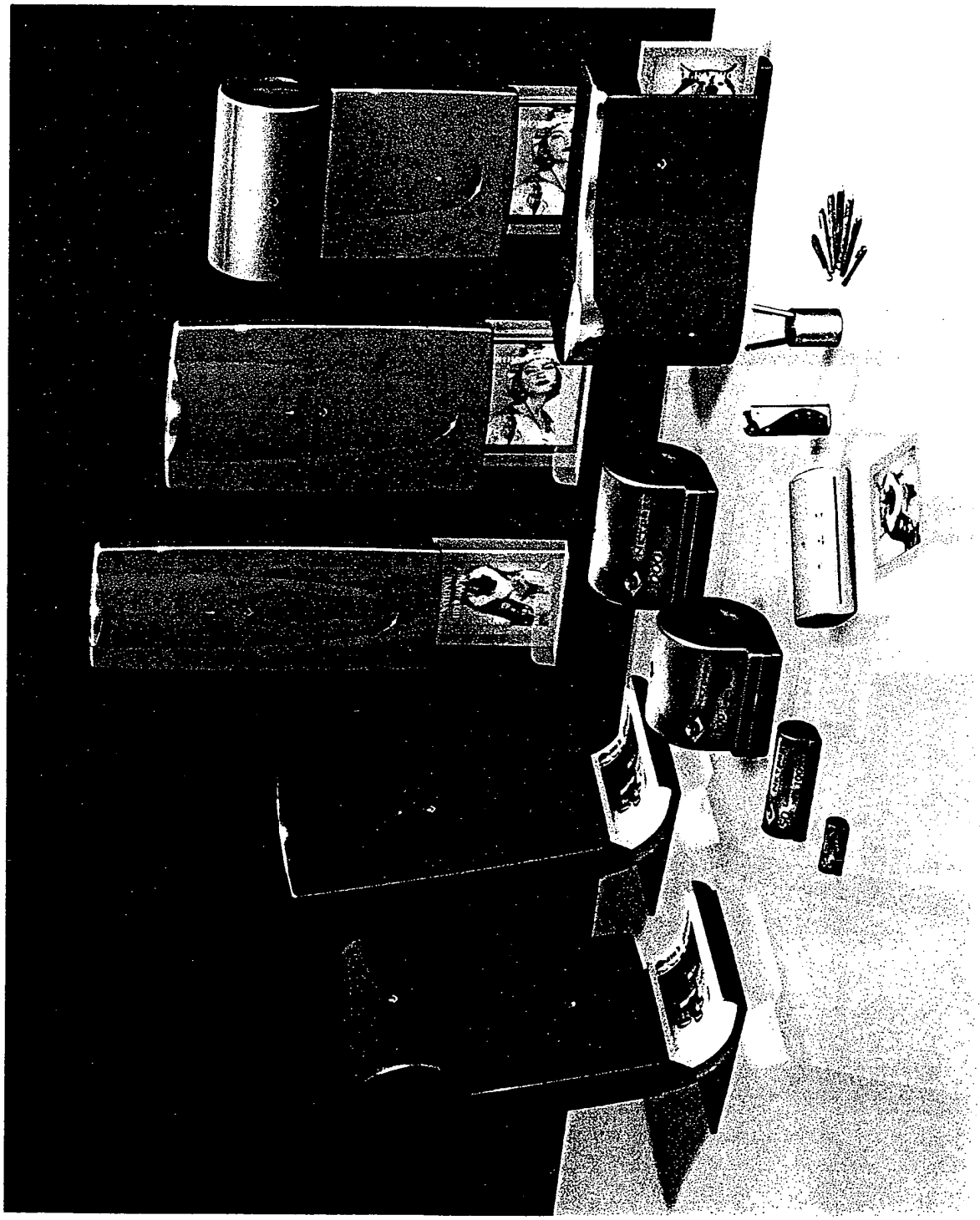


Figure 2. Netpage Printer family

5.5 THE NETPAGE PEN

The Netpage Pen operates both as a normal marking ink pen and as a non-marking stylus. When either nib is in contact with a Netpage, the pen continuously monitors its movements relative to the page. The nib is attached to a pressure sensor. The pen pressure can be interpreted relative to a threshold to indicate whether the pen is "up" or "down". It can also be interpreted as a continuous value, for example when the pen is capturing a signature, to allow the full dynamics of the signature to be verified.

The pen determines the position of its nib on the Netpage by imaging, in the infrared spectrum, an area of the page in the vicinity of the nib. It decodes the nearest page id and position tag, and adjusts the position given by the tag to account for the distance between the area imaged and the actual nib, and the position of the tag in the imaged area. Although the position resolution of the tag may be low, because the tag density on the page is inversely proportional to the tag size, the adjusted position resolution is quite high, and easily exceeds the minimum 200 dpi resolution required for handwriting recognition [80].

Pen actions relative to a Netpage consist of a series of strokes. A stroke consists of a sequence of time-stamped pen positions on the page, initiated by a pen-down event and completed by the subsequent pen-up event. A stroke is also tagged with the page id of the Netpage whenever the page id changes, i.e. just at the start of the stroke under normal circumstances.

The position tags on the Netpage contain various control bits. One of these instructs the pen to activate its "active area" LED. Thus a region on the page which corresponds to the active area of a button or hyperlink can be encoded to activate this LED, giving the user visual feedback that the button or hyperlink is active when the pen passes over it. Another control bit instructs the pen to capture continuous pen pressure readings and tag the stroke with these readings. Thus a region on the page which corresponds to a signature input area can be encoded to capture continuous pen pressure.

Whenever the pen is within range of a printer with which it can communicate, the pen slowly flashes its "online" LED. When the pen fails to decode a stroke relative to the page, it momentarily activates its "error" LED. When the pen succeeds in decoding a stroke relative to the page, it momentarily activates its "ok" LED.

The pen also contains a pair of passive accelerometers mounted at right angles to each other in the plane normal to the pen's axis. The accelerometers respond to gravity and allow the pen to compute its tilt. This in turn helps it auto-focus its optics and compute the nib-to-tag displacement. If the stroke is being tagged with pen pressure readings, then it is also tagged with tilt readings.

A sequence of captured strokes, whether tagged with pen pressure and tilt or not, is referred to as *digital ink*. Digital ink forms the basis for the digital exchange of drawings and handwriting, for on-line recognition of handwriting [80], and for on-line verification of signatures.

The pen is wireless and transmits digital ink to the Netpage Printer using a radio frequency signal. The digital ink data is encrypted for security and packetized for efficient transmission, but is always flushed on a pen-up event to ensure timely handling in the printer.

When the pen is out-of-range of a printer it buffers digital ink in internal memory, which has a capacity of more than 12 minutes of continuous handwriting. When the pen is once again within range of a printer, it transfers any buffered digital ink.

A pen can be registered with any number of printers, but because all state data resides in Netpages both on paper and on the network, it is largely immaterial which printer a pen is communicating with at any particular time.

5.6 NETPAGE INTERACTION

When the Netpage Printer receives a digital ink stroke from the pen, it retrieves the page layout of the Netpage identified in the stroke, to allow it to correctly interpret the stroke. The printer resolves, via the DNS, the address of the Netpage Page Server which holds the page layout, and then retrieves the page layout from the server. If the page was recently identified in an earlier stroke, then the printer may already have the address of the relevant Netpage Page Server in its cache. It may also have the page layout itself in its cache, in which case there may be no need to retrieve it.

Once the printer has the page layout of the Netpage to which the pen stroke refers, it can interpret the stroke in relation to the layout and content of the page. This involves hit-testing the objects on the page to determine which objects the pen is interacting with, in much the same way that mouse movements and button presses are interpreted in a graphical user interface system.

A "click" is a stroke where the distance between the pen down position and the subsequent pen up position is less than some small maximum. An object which is activated by a click requires a click to be activated, i.e. a longer stroke is ignored. The failure of a pen action, such as a "sloppy" click, to register is indicated by the lack of response from the pen's "ok" LED.

There are two kinds of interactive objects on a Netpage: hyperlinks and form fields.

When a hyperlink is activated, the printer sends a request to a handler somewhere on the network. The handler is identified by a URI, and the URI is resolved in the normal way via the DNS. There are three types of hyperlinks: general hyperlinks, form hyperlinks, and selection hyperlinks. A general hyperlink may implement a request for a linked document, or may simply signal a preference to a server. A form hyperlink submits the corresponding form to a form handler. A selection hyperlink submits the current selection to a selection handler. If the current selection contains a single-word piece of text, for example, the selection handler may return a single-page document giving the word's meaning within the context in which it appears, or a translation into a different language. Each hyperlink type is characterized by what information is submitted to the handler.

Form fields come in four varieties: checkboxes, text areas, digital ink areas, and signature areas. A checkbox accepts a true or false value. Any mark (a tick, a cross, a stroke, a fill zigzag, etc.) captured in a checkbox area is assigned as a true value to the corresponding field. A text area accepts a text string. Any digital ink captured in a text area is automatically converted to text via on-line handwriting recognition and the text is assigned to the corresponding field. A digital ink area accepts raw digital ink. Any digital ink captured in a digital ink area is assigned to the corresponding field. A signature area accepts a handwritten signature. Any digital ink captured in a signature area is automatically verified and the resulting signature token is assigned to the corresponding signature field. Signature verification is discussed in more detail in Section 8.

"Editing" commands, such as strike-throughs indicating deletion, are also recognized in form fields.

Table 2. Summary of pen interactions with a Netpage

object	type	pen input	action
hyperlink	general	click	submit action to handler via URI
	form	click	submit form to handler via URI
	selection	click	submit selection to handler via URI
form field	checkbox	any mark	set field value to true
	text area	handwriting	convert digital ink to text; assign text to field
	digital ink area	digital ink	assign digital ink to field
	signature area	signature	verify digital ink signature; assign signature token to field
none	-	circumscription	convert digital ink to region; select object(s) in region

Because the handwriting recognition algorithm works "on-line" (i.e. with access to the dynamics of the pen movement), rather than "off-line" (i.e. with access only to a bitmap of pen markings), it can recognize run-on discretely-written characters [80] with high accuracy, without a writer-dependent training phase.

Digital ink, as already stated, consists of a sequence of strokes. Any stroke which starts in a particular object's active area is appended to that area's digital ink stream, ready for eventual interpretation. Any stroke not appended to an object's digital ink stream is appended to the remaining inactive area's digital ink stream.

Digital ink captured in the inactive area is interpreted as a selection gesture. Any circumscription of one or more objects is interpreted as a selection of the circumscribed objects.

The printer maintains a current selection for each pen. The selection contains the most recent object selected, resolved with reference to the page layout and content. The selection can be attached to or pasted into another form, or in general be submitted to a selection handler as described earlier. The selection is cleared after an inactivity time-out to ensure predictable behavior.

Table 2 provides a summary of pen interactions with a Netpage.

5.7 FORMS

As described in Section 5.2, user input on a physical Netpage is ultimately recorded persistently by a Netpage Page Server together with the corresponding page instance. To ensure efficient capture of user input, the printer accumulates input locally. To prevent update anomalies, however, the printer temporarily obtains exclusive access to the page instance from the Netpage Page Server. The printer flushes input back to the server and relinquishes exclusive access when the user initiates a non-local action on the page; after an inactivity time-out on the page; when the printer wishes to free up local storage consumed by the page; and on request from the server.

When the printer submits a form to a form handler, it simply submits the document instance of the form. The form handler retrieves the field values from the Netpage Page Server at its leisure.

A form can also act as a shared "blackboard" between the user and the form handler, i.e. the form handler can query the contents of the form fields maintained by the Netpage Page Server without the user explicitly submitting the form.

For text areas, the raw digital ink is optionally also stored with the page instance on the Netpage Page Server. This allows the form handler to interrogate the raw digital ink should it suspect the original recognition of the handwriting. This might involve human intervention at the application level for forms which fail certain application-specific consistency checks. As an extension to this, the entire background area of a form can be designated as a digital ink area. The form handler can then decide, on the basis of the presence of digital ink outside the explicit fields of the form, to route the form to a human operator, on the assumption that the user may have indicated amendments to the filled-in fields outside of those fields.

Form fields can optionally be tagged to indicate their meaning. Fields tagged in this way may include name and address fields, for example. This semantic tagging allows these fields to be automatically filled in whenever a "blank" form is requested by an identifiable user, i.e. a user who has registered their identity with the system and linked it to the identity of their pen.

5.8 STANDARD FEATURES OF NETPAGES

Each Netpage is printed with the Netpage logo at the bottom to indicate that it is a Netpage and therefore has interactive properties. The logo also acts as a "copy" button. In most cases pressing the logo produces a copy of the page. In the case of a form the button instead elicits a page giving the user the option to print the entire form document. And in the case of a secure document, such as a ticket or coupon, the button elicits an explanatory note or advertising page.

The default single-page copy function is handled directly by the relevant Netpage Page Server. Special copy functions are handler by linking the logo button to other URIs.

Once a Netpage form has been submitted, it is marked as submitted by the Netpage Page Server and cannot be submitted again. An attempt to do so elicits a status report indicating when it was submitted. A copy of the form can still be made, altered, and re-submitted.

5.9 THE HELP SYSTEM

The Netpage Printer has a single button labelled "help". When pressed it elicits a single page of information. This information includes the following:

- status of printer connection
- status of printer consumables
- top-level help menu
- document function menu
- top-level Netpage Network directory

The help menu provides a hierarchical manual on how to use the Netpage System.

The document function menu includes the following functions:

- print a copy of a document

- print a clean copy of a form
- print the status of a document

A document function is initiated by simply pressing the button and then touching any page of the document. The status of a document indicates who published it and when, to whom it was delivered, and to whom and when it was subsequently submitted as a form.

The Netpage Network directory allows the user to navigate the hierarchy of publications and services on the network. As an alternative, the user can call the Netpage Network "900" number "yellow pages" and speak to a human operator. The operator can locate the desired document and route it to the user's printer. Depending on the document type, the publisher or the user pays the small "yellow pages" service fee.

The help page is obviously unavailable if the printer is unable to print. In this case the "error" light is lit and the user can request remote diagnosis over the network.

6 Personalized Publication Model

In the following discussion, news is used as a canonical publication example to illustrate personalization mechanisms in the Netpage System. Although news is often used in the limited sense of newspaper and news magazine news, the intended scope is wider.

In the Netpage System, the editorial content and the advertising content of a news publication are personalized using different mechanisms. The editorial content is personalized according to the reader's explicitly stated and implicitly captured interest profile. The advertising content is personalized according to the reader's locality and demographic.

6.1 EDITORIAL PERSONALIZATION

A subscriber can draw on two kinds of news sources: those that deliver news publications, and those that deliver news streams. While news publications are aggregated and edited by the publisher, news streams are aggregated either by a news publisher or by a specialized news aggregator. News publications typically correspond to traditional newspapers and news magazines, while news streams can be many and varied: a "raw" news feed from a news service, a cartoon strip, a freelance writer's column, a friend's bulletin board, or the reader's own e-mail.

The Netpage Publication Server supports the publication of edited news publications as well as the aggregation of multiple news streams. By handling the aggregation and hence the formatting of news streams selected directly by the reader, the server is able to place advertising on pages over which it otherwise has no editorial control.

The subscriber builds a daily newspaper by selecting one or more contributing news publications, and creating a personalized version of each. The resulting daily editions are printed and bound together into a single newspaper. The various members of a household typically express their different interests and tastes by selecting different daily publications and then customizing them.

For each publication, the reader optionally selects specific sections. Some sections appear daily, while others appear weekly. The daily sections available from The New York Times online, for example, include "Page One Plus", "National", "International", "Opinion", "Business", "Arts/Living", "Technology", and "Sports". The set of available sections is obviously specific to a publication, as is the default subset.

The reader extends the daily newspaper by creating custom sections, each one drawing on any number of news streams. Custom sections might be created for e-mail and friends' announcements ("Personal"), or for monitoring news feeds for specific topics ("Alerts" or "Clippings").

For each section, the reader optionally specifies its size, either qualitatively (e.g. short, medium, or long), or numerically (i.e. as a limit on its number of pages), and the desired proportion of advertising, either qualitatively (e.g. high, normal, low, none), or numerically (i.e. as a percentage).

The reader also optionally expresses a preference for a large number of shorter articles or a small number of longer articles. Each article is ideally written (or edited) in both short and long forms to support this preference.

An article may also be written (or edited) in different versions to match the expected sophistication of the reader, for example to provide children's and adults' versions. The appropriate version is selected according to the reader's age. The reader can specify a "reading age" which takes precedence over their biological age.

The articles which make up each section are selected and prioritized by the editors, and each is assigned a useful lifetime. By default they are delivered to all relevant subscribers, in priority order, subject to space constraints in the subscribers' editions.

In sections where it is appropriate, the reader may optionally enable collaborative filtering. This is then applied to articles which have a sufficiently long lifetime. Each article which qualifies for collaborative filtering is printed with rating buttons at the end of the article. The buttons can provide an easy choice (e.g. "liked" and "disliked'), making it more likely that readers will bother to rate the article.

Articles with high priorities and short lifetimes are therefore effectively considered essential reading by the editors and are delivered to most relevant subscribers.

The reader optionally specifies a serendipity factor, either qualitatively (e.g. do or don't surprise me), or numerically. A high serendipity factor lowers the threshold used for matching during collaborative filtering. A high factor makes it more likely that the corresponding section will be filled to the reader's specified capacity. A different serendipity factor can be specified for different days of the week.

The reader also optionally specifies topics of particular interest within a section, and this modifies the priorities assigned by the editors.

The speed of the reader's Internet connection affects the quality at which images can be delivered. The reader optionally specifies a preference for fewer images or smaller images or both. If the number or size of images is not reduced, then images may be delivered at lower quality (i.e. at lower resolution or with greater compression).

At a global level, the reader specifies how quantities, dates, times and monetary values are localized. This involves specifying whether units are imperial or metric, a local timezone and time format, and a local currency, and whether the localization consist of *in situ* translation or annotation. These preferences are derived from the reader's locality by default.

To reduce reading difficulties caused by poor eyesight, the reader optionally specifies a global preference for a larger presentation. Both text and images are scaled accordingly, and less information is accommodated on each page.

The language in which a news publication is published, and its corresponding text encoding, is a property of the publication and not a preference expressed by the user. However, the Netpage Network may provide automatic translation services in various guises.

6.2 Advertising Localization and Targeting

The personalization of the editorial content directly affects the advertising content, because advertising is typically placed to exploit the editorial context. Travel ads, for example, are more likely to appear in a travel section than elsewhere. The value of the editorial content to an advertiser (and therefore to the publisher) lies in its ability to attract large numbers of readers with the right demographics.

Effective advertising is placed on the basis of locality and demographics. Locality determines proximity to particular services, retailers etc., and particular interests and concerns associated with the local community and environment. Demographics determine general interests and preoccupations as well as likely spending patterns.

A news publisher's most profitable product is advertising "space", a multi-dimensional entity determined by the publication's geographic coverage, the size of its readership, its readership demographics, and the page area available for advertising.

In the Netpage System, the Netpage Publication Server computes the approximate multi-dimensional size of a publication's saleable advertising space on a per-section basis, taking into account the publication's geographic coverage, the section's readership, the size of each reader's section edition, each reader's advertising proportion, and each reader's demographic.

In comparison with other media, the Netpage System allows the advertising space to be defined in greater detail, and allows smaller pieces of it to be sold separately. It therefore allows it to be sold at closer to its true value.

For example, the same advertising "slot" can be sold in varying proportions to several advertisers, with individual readers' pages randomly receiving the advertisement of one advertiser or another, overall preserving the proportion of space sold to each advertiser.

The Netpage System allows advertising to be linked directly to detailed product information and online purchasing. It therefore raises the intrinsic value of the advertising space.

Because personalization and localization are handled automatically by Netpage Publication Servers, an advertising aggregator can provide arbitrarily broad coverage of both geography and demographics. The subsequent disaggregation is efficient because it is automatic. This makes it more cost-effective for publishers to deal with advertising aggregators than to directly capture advertising. Even though the advertising aggregator is taking a proportion of advertising revenue, publishers may find the change profit-neutral because of the greater efficiency of aggregation. The advertising aggregator acts as an intermediary between advertisers and publishers, and may place the same advertisement in multiple publications.

It is worth noting that ad placement in a Netpage publication can be more complex than ad placement in the publication's traditional counterpart, because the publication's advertising space is more complex. While ignoring the full complexities of negotiations between advertisers, advertising aggregators and publishers, it is clear that the Netpage System should ideally provide some automated support for these negotiations, including support for automated auctions of advertising space. Automation is particularly desirable for the placement of advertisements which generate small amounts of income, i.e. small or highly localized advertisements.

Once placement has been negotiated, the aggregator captures and edits the advertisement and records it on a Netpage Ad Server. Correspondingly, the publisher records the ad placement on the relevant Netpage Publication Server. When the Netpage Publication Server lays out each user's personalized publication, it picks the relevant advertisements from the Netpage Ad Server.

6.3 USER PROFILES

The personalization of news and other publications relies on an assortment of user-specific profile information:

- publication customizations
- collaborative filtering vectors
- contact details
- presentation preferences

The customization of a publication is typically publication-specific, and so the customization information is maintained by the relevant Netpage Publication Server.

A collaborative filtering vector consists of the user's ratings of a number of news items. As described in Section 4, it is used to correlate different users' interests for the purposes of making recommendations. Although there are benefits to maintaining a single collaborative filtering vector independently of any particular publication, there are two reasons why it is more practical to maintain a separate vector for each publication: there is likely to be more overlap between the vectors of subscribers to the same publication than to different publications; and a publication is likely to want to present its users' collaborative filtering vectors as part of the value of its brand, not to be found elsewhere. Collaborative filtering vectors are therefore also maintained by the relevant Netpage Publication Server.

Contact details, including name, street address, ZIP code, state, country, telephone numbers, etc., are by their nature global and are maintained by a Netpage Registration Server.

Presentation preferences, including those for quantities, dates and times discussed in Section 6.1, are likewise global and maintained in the same way.

The localization of advertising relies on the locality indicated in the user's contact details, while the targeting of advertising relies on personal information such as date of birth, gender, marital status, income, profession, education, etc., or qualitative derivatives such as age range and income range.

For those users who choose to reveal personal information for advertising purposes, the information is maintained by the relevant Netpage Registration Server. In the absence of such information, advertising can be targeted on the basis of the demographic associated with the user's ZIP or ZIP+4 code.

Each user, pen and printer is assigned a globally unique identifier, and the Netpage Registration Server maintains the relationships between them. The server also keeps track of which publications a user has authorized to print on particular printers. Each user may have several pens, but a pen is specific to a single user. A pen may know any number of printers, and a printer may know any number of pens. These relationships are illustrated in Figure 3.

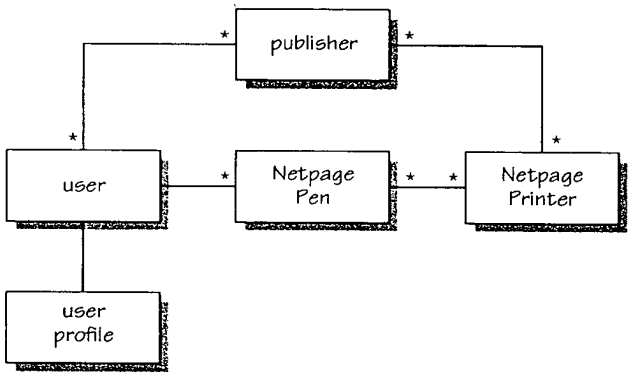


Figure 3. User registration relationships

The pen identifier is used, in the form of a URN, to locate the corresponding user profile maintained by a particular Netpage Registration Server, via the DNS in the usual way.

6.4 INTELLIGENT PAGE LAYOUT

The Netpage Publication Server automatically lays out the pages of each user's personalized publication on a section-by-section basis. Since most advertisements are in the form of pre-formatted rectangles, they are placed on the page before the editorial content.

The advertising ratio for a section can be achieved with wildly varying advertising ratios on individual pages within the section, and the ad layout algorithm exploits this. The algorithm attempts to co-locate closely tied editorial and advertising content, e.g. ads for roofing material placed specifically with the publication because of a special feature on do-it-yourself roofing repairs.

The editorial content selected for the user, i.e. text with associated images and graphics, is then laid out according to various aesthetic rules.

The entire process, including the selection of ads and the selection of editorial content, must be iterated once the layout has converged, to attempt to more closely achieve the user's stated section size preference. The section size preference can, however, be matched *on average* over time, allowing significant day-to-day variations.

6.5 DOCUMENT FORMAT

Once the document is laid out, it is encoded for efficient distribution and persistent storage on the Netpage Network.

The primary efficiency mechanism is the separation of information specific to a single user's edition and information shared between multiple users' editions. The specific information consists of the page layout. The shared information consists of the objects to which the page layout refers, including images, graphics, and pieces of text.

A text object contains fully-formatted text represented in the Extensible Markup Language (XML) [92] using the Extensible Stylesheet Language (XSL) [93]. XSL provides precise control over text formatting independently of the region into which the text is

being set, which in this case is being provided by the layout. The text object contains embedded language codes to enable automatic translation, and embedded hyphenation hints to aid with paragraph formatting.

An image object encodes an image in the JPEG 2000 wavelet-based compressed image format [46]. The original DCT-based JPEG algorithm introduces negligible visual loss at compression ratios below 10:1 [88]. JPEG 2000 is planned to achieve the same quality at compression ratios 30% higher, i.e. at about 13:1 [47].

A graphic object encodes a 2D graphic in Scalable Vector Graphics (SVG) [95] format.

The layout itself consists of a series of placed image and graphic objects, linked textflow objects through which text objects flow, hyperlinks and input fields as described in Section 5.6, and watermark regions. These layout objects are summarized in Table 3. The layout uses a compact format suitable for efficient distribution and storage.

The layout is tagged with the version of the text-setting algorithm used by the Netpage Publication Server when the layout was first created, allowing the Netpage Printer to exactly reproduce the physical layout intended by the server.

Because Netpage Printer software is automatically upgraded over the Netpage Network, it is feasible to for Netpage Printers to contain every version of the text-setting algorithm.

Table 3. Netpage layout objects

layout object	attribute	linked object format
image	position	-
	image object URI	JPEG 2000
graphic	position	-
	graphic object URI	SVG
textflow	textflow id	-
	region[a]	
	optional text object URI	XML/XSL
hyperlink	type	-
	region[a]	-
	handler URI	-
field	type	-
	meaning	
	region[a]	-
watermark	region[a]	-

a. arbitrary multi-edged shape defined with spline paths

6.6 DOCUMENT DISTRIBUTION

As described above, for purposes of efficient distribution and persistent storage on the Netpage Network, a user-specific page layout is separated from the shared objects to which it refers.

When a subscribed publication is ready to be distributed, the Netpage Publication Server allocates, with the help of the Netpage Id Server, a globally unique id for each page, page instance, document, and document instance.

The server computes a set of optimized subsets of the shared content and creates a multicast channel for each subset, and then tags each user-specific layout with the names of the multicast channels which will carry the shared content used by that layout. The server then pointcasts each user's layouts to that user, and when the pointcasting is complete, multicasts the shared content on the specified channels. After receiving its pointcast, each Netpage Printer subscribes to the multicast channels specified in the page layouts. During the multicasts, each printer extracts from the multicast streams those objects referenced by its page layouts.

Once a printer has received all the objects to which it page layouts refer, the printer re-creates the fully-populated layout and then rasterizes and prints it.

The server also delivers each page layout to the relevant Netpage Page Server, which may be co-located with the Netpage Publication Server, or may be located elsewhere on the network. Thus the page layouts are persistently archived as Netpages. It is the responsibility of the Netpage Publication Server to preserve the shared objects referenced by the Netpages, to ensure that they are really persistent. It may choose to archive these shared objects elsewhere on the network at any time. The object URIs embedded in the Netpages allow the objects to move.

Under normal circumstances, the printer prints page faster than they can be delivered. Assuming a quarter of each page is covered with images, the average page has a size of less than 400KB[1]. The printer can therefore hold in excess of 100 such pages in its internal 64MB memory, allowing for temporary buffers etc. The printer prints at a rate of one page per second. This is equivalent to 400KB or about 3Mbit of page data per second, which is similar to the highest expected rate of page data delivery over a broadband network.

Even under abnormal circumstances, such as when the printer runs out of paper, it is likely that the user will be able to replenish the paper supply before the printer's 100-page internal storage capacity is exhausted.

However, if the printer's internal memory does fill up, then the printer will be unable to make use of a multicast when it first occurs. The Netpage Publication Server therefore allows printers to submit requests for re-multicasts. When a critical number of requests is received or a timeout occurs, the server re-multicasts the corresponding shared objects.

Once a document is printed, a Netpage Printer can produce an exact duplicate at any time by retrieving its page layouts from the relevant Netpage Page Server and retrieving the objects to which they refer from the network.

6.7 On-Demand Documents

When a Netpage document is requested ad hoc, it is personalized and delivered in much the same way as a periodical. However, since there is no shared content, delivery is made directly to the requesting printer, i.e. without the use of multicast.

1. 267 pixels per inch (ppi) 24-bit RGB, compressed using JPEG 2000 at a ratio of 13:1.

When a non-Netpage document is requested ad hoc, it is not personalized, and it is delivered via a designated Netpage Formatting Server which reformats it as a Netpage document. A Netpage Formatting Server is a special instance of a Netpage Publication Server. The Netpage Formatting Server has knowledge of myriad Internet document formats, including old favorites such as Adobe's Portable Document Format (PDF) [5], and Hypertext Markup Language (HTML) [94]. In the case of HTML, it makes use of the higher resolution of the printed page to present Web pages in a two-column format, with a table of contents and an index of links. By default it automatically includes all Web pages directly linked to the requested page. The user can tune this behavior via a preference.

The Netpage Formatting Server makes standard Netpage behavior, including interactivity and persistence, available on any Internet document, no matter what its origin and format. It hides knowledge of different document formats from both the Netpage Printer and the Netpage Page Server.

7 Security

7.1 CRYPTOGRAPHY

Cryptography is used to protect sensitive information, both in storage and in transit, and to authenticate parties to a transaction. There are two classes of cryptography in widespread use: secret-key cryptography and public-key cryptography. The Netpage Network uses both classes of cryptography.

Secret-key cryptography, also referred to as symmetric cryptography, uses the same key to encrypt and decrypt a message. Two parties wishing to exchange messages must first arrange to securely exchange the secret key.

Public-key cryptography, also referred to as asymmetric cryptography, uses of two encryption keys. The two keys are mathematically related in such a way that any message encrypted using one key can only be decrypted using the other key. One of these keys is then published, while the other is kept private. The public key is used to encrypt any message intended for the holder of the private key. Once encrypted using the public key, a message can only be decrypted using the private key. Thus two parties can securely exchange messages without first having to exchange a secret key. To ensure that the private key is secure, it is normal for the holder of the private key to generate the key pair.

Public-key cryptography can be used to create a digital signature. If the holder of the private key creates a known hash of a message and then encrypts the hash using the private key, then anyone can verify that the encrypted hash constitutes the "signature" of the holder of the private key with respect to that particular message, simply by decrypting the encrypted hash using the public key and verifying the hash against the message. If the signature is appended to the message, then the recipient of the message can verify both that the message is genuine and that it has not been altered in transit.

To make public-key cryptography work, there has to be a way to distribute public keys which prevents impersonation. This is normally done using certificates and certificate authorities. A certificate authority is a trusted third party which authenticates the connection between a public key and someone's identity. The certificate authority verifies the person's identity by examining identity documents etc., and then creates and signs a digital certificate containing the person's identity details and public key. Anyone who trusts the certificate authority can use the public key in the certificate with a high degree of certainty that it is genuine. They just have to verify that the certificate has indeed been signed by the certificate authority, whose public key is well-known.

In most transaction environments, public-key cryptography is only used to create digital signatures and to securely exchange secret session keys. Secret-key cryptography is used for all other purposes.

In the following discussion, when reference is made to the *secure* transmission of information between a Netpage Printer and a server, what actually happens is that the printer obtains the server's certificate, authenticates it with reference to the certificate authority, uses the public key-exchange key in the certificate to exchange a secret session key with the server, and then uses the secret session key to encrypt the message data. A *session* key, by definition, can have an arbitrarily short lifetime.

7.2 NETPAGE PRINTER SECURITY

Each Netpage Printer is assigned a pair of unique identifiers at time of manufacture which are stored in read-only memory in the printer and in the Netpage Registration Server database. The first id is public and uniquely identifies the printer on the Netpage Network. The second id is secret and is used when the printer is first registered on the network.

When the printer connects to the Netpage Network for the first time after installation, it creates a signature public/private key pair. It transmits the secret id and the public key securely to the Netpage Registration Server. The server compares the secret id against the printer's secret id recorded in its database, and accepts the registration if the ids match. It then creates and signs a certificate containing the printer's public id and public signature key, and stores the certificate in the registration database.

The Netpage Registration Server acts as a certificate authority for Netpage Printers, since it has access to secret information allowing it to verify printer identity.

When a user subscribes to a publication, a record is created in the Netpage Registration Server database authorizing the publisher to print on a particular printer. Every document sent to a printer is signed by the publisher using the publisher's private signature key. The printer verifies via the registration database that the publisher is authorized to print on the printer, and verifies the signature using the publication's public key, obtained from the publisher's certificate stored in the registration database.

The Netpage Registration Server accepts requests to add printing authorizations to the database, so long as those requests are initiated via a pen registered to the printer.

7.2.1 Casual Printing Authorizations

The user can register a Web terminal as a "publisher" authorized to print on a printer. This is useful if the user has a Web terminal in the home which is used to locate documents on the Web for printing. The one-time authorization proceeds as follows: the user prints a Web terminal authorization form. The Netpage Registration Server generates a short-lifetime one-time-use id for the Web terminal which is printed on the form, together with the URI of the printer. The Web terminal is used to navigate to a Netpage Registration Server registration site, where the one-time-use id is entered, as well as the URI of the printer. The Web terminal generates a signature public/private key pair. The server allocates a publisher id for the Web terminal, creates and signs a certificate containing the publisher id and the public key, and stores the certificate in the registration database. The URI of the printer, the Web terminal's publisher id, and the private signature key are stored locally on the Web terminal.

Whenever the Web terminal wishes to print on the printer, it sends the printer's designated Netpage Formatting Server a request containing the URI of the document to be printed, together with the publisher id, signed with the Web terminal's private signature key. On receipt of the request and before acting on it, the server verifies the publisher id and signature in the usual way.

The user can print at list of current printing authorizations at any time, and revoke any which are being abused.

The same scheme can be used to authorize other Netpage users to send greeting cards, e-mail, etc., to the printer. The user simply prints a one-time authorization "token". This is given to the other user, who simply presses the button on the token with their pen. The

system takes care of the rest, including adding each user's name to the other's address book.

Authorization tokens can be printed on the back of a user's business card, to provide casual but fairly controlled authorization. Recall that authorizations can be easily reviewed and selectively revoked.

A user can also choose to provide all users with greeting card, e-mail, etc., access to their Netpage Printer.

7.3 NETPAGE PEN SECURITY

Each Netpage Pen is assigned a unique identifier at time of manufacture which is stored in read-only memory in the pen and in the Netpage Registration Server database. The id uniquely identifies the pen on the Netpage Network.

A Netpage Pen can know a number of Netpage Printers, and a printer can know a number of pens. A pen communicates with a printer via a radio frequency signal whenever it is within range of the printer. Once a pen and printer are registered, they regularly exchange session keys. Whenever the pen transmits digital ink to the printer, it always uses the appropriate session key. Digital ink is never transmitted in the clear.

A pen stores a session key for every printer is knows, indexed by printer id, and a printer stores a session key for every pen it knows, indexed by pen id. Both have a large but finite storage capacity for session keys, and will forget a session key on a least-recently-used basis if necessary. If either a pen or a printer forgets the other, then they simply have to go though the automatic registration procedure again.

When an unknown pen comes within range of a printer, they soon discover they don't know each other. Under these circumstances the pen simply ignores the printer until it finds itself in the charging cup, at which time it initiates the registration procedure.

In addition to its public id, the pen contains a secret id and a secret key-exchange key, both intended for one-time use. These are also recorded in the Netpage Registration Server database at time of manufacture. During registration, the printer obtains the secret id from the pen. Because it is transmitted in the clear, it may be intercepted by someone listening in. The printer transmits the id securely to the Netpage Registration Server, which responds securely with the matching key-exchange key, together with a newly-generated secret id and key-exchange key. The printer generates a session key for the pen and transmits it to the pen encrypted using the one-time-use key-exchange key. It also securely transmits the new secret id and key-exchange key to the pen, which saves them for the next registration procedure. They now match the pen's record in the Netpage Registration Server database.

If the secret id transmitted in the clear from the pen to the printer is intercepted and used to retrieve the secret key-exchange key from the Netpage Registration Server before the printer queries the server, then the server rejects the printer's query because the secret id is out-of-date. Thus the printer knows that the pen has been compromised, and recommends that it be returned for repair.

Whenever a pen is registered, the Netpage Registration Server prints a registration form allowing the pen to be registered in the name of a user. The pen can be registered to an

existing user in the registration database, in which case the user's optional password may have to be provided, or new user details can be entered.

The pen uses secret-key rather than public-key encryption because of hardware performance constraints in the pen.

7.4 SECURE DOCUMENTS

The Netpage System supports the delivery of secure documents such as tickets and coupons. The Netpage Printer includes a facility to print watermarks, but will only do so on request from publishers who are suitably authorized. The publisher indicates its authority to print watermarks in its certificate, which the printer is able to authenticate.

The "watermark" printing process uses an alternative dither matrix in specified "watermark" regions of the page. Back-to-back pages contain mirror-image watermark regions which coincide when printed. The dither matrices used in odd and even pages' watermark regions are designed to produce an interference effect when the regions are viewed together - i.e. when looking *through* the printed sheet.

The effect is similar to a watermark in that it is not visible when looking at only one side of the page, and is lost when the page is copied by normal means.

As described in Section 5.8, pages of secure documents cannot be copied using the built-in Netpage copy mechanism. This extends to copying Netpages on Netpage-aware photocopiers.

Secure documents are typically generated as part of e-commerce transactions. They can therefore include the user's photograph which was captured when the user registered biometric information with the Netpage Registration Server, as described in Section 8.

When presented with a secure Netpage document, the recipient can verify its authenticity by requesting its status in the usual way. The unique id of a secure document is only valid for the lifetime of the document, and secure document ids are allocated non-contiguously to prevent their prediction by opportunistic forgers. A secure document verification pen can be developed with built-in feedback on verification failure, to support easy point-of-presentation document verification.

Clearly neither the watermark nor the user's photograph are secure in a cryptographic sense. They simply provide a significant obstacle to casual forgery. Online document verification, particularly using a verification pen, provides an added level of security where it is needed, but is still not entirely immune to forgeries.

7.5 NON-REPUDIATION

In the Netpage System, forms submitted by users are delivered reliably to forms handlers and are persistently archived on Netpage Page Servers. It is therefore impossible for recipients to repudiate delivery.

E-commerce payments made through the system, as described in Section 8, are also impossible for the payee to repudiate.

8 Electronic Commerce Model

8.1 Secure Electronic Transaction (SET)

The Netpage System uses the Secure Electronic Transaction (SET) [58] system as its payment system model. Although SET is not yet widely supported, it is comprehensive and elegant and will probably become dominant in the near future.

SET, having been developed by MasterCard and Visa, is organized around payment cards, and this is reflected in the terminology. However, much of the system is independent of the type of accounts being used.

In SET, cardholders and merchants register with a certificate authority and are issued with certificates containing their public signature keys. The certificate authority verifies a cardholder's registration details with the card issuer as appropriate, and verifies a merchant's registration details with the acquirer as appropriate. Cardholders and merchants store their respective private signature keys securely on their computers. During the payment process, these certificates are used to mutually authenticate a merchant and cardholder, and to authenticate them both to the payment gateway.

SET has not yet been adopted widely, partly because cardholder maintenance of keys and certificates is considered burdensome. Interim solutions which maintain cardholder keys and certificates on a server and give the cardholder access via a password have met with some success [18].

8.2 SET Payments

In the Netpage System the Netpage Registration Server acts as a proxy for the Netpage user (i.e. the cardholder) in SET payment transactions.

The Netpage System uses biometrics to authenticate the user and authorize SET payments. Because the system is pen-based, the biometric used is the user's on-line signature, consisting of time-varying pen position, tilt and pressure. A fingerprint biometric can also be used by designing a fingerprint sensor into the pen, although at a higher cost. The type of biometric used only affects the capture of the biometric, not the authorization aspects of the system.

The first step to being able to make SET payments is to register the user's biometric with the Netpage Registration Server. This is done in a controlled environment, for example a bank, where the biometric can be captured at the same time as the user's identity is verified. The biometric is captured and stored in the registration database, linked to the user's record and to the record of a particular Netpage Pen. The user's photograph is also optionally captured and linked to the record. The SET cardholder registration process is completed, and the resulting private signature key and certificate are stored in the database. The user's payment card information is also stored, giving the Netpage Registration Server enough information to act as the user's proxy in any SET payment transaction.

When the user eventually supplies the biometric to complete a payment, for example by signing a Netpage order form, the printer securely transmits the order information, the pen id and the biometric data to the Netpage Registration Server. The server verifies the biometric with respect to the user identified by the pen id, and from then on acts as the user's proxy in completing the SET payment transaction.

8.3 MICRO-PAYMENTS

The Netpage Network includes a mechanism for micro-payments, to allow the user to be conveniently charged for printing low-cost documents on demand and for copying copyright documents, and possibly also to allow the user to reimbursed for expenses incurred in printing advertising material. The latter depends on the level of subsidy already provided to the user.

When the user registers for e-commerce, a network account is established which aggregates micro-payments. The user receives a statement on a regular basis, and can settle any outstanding debit balance using the standard payment mechanism.

The network account can be extended to aggregate subscription fees for periodicals, which would also otherwise be presented to the user in the form of individual statements.

8.4 TRANSACTIONS

Whenever a transaction originates through a Netpage form, the form handler has sufficient information, in the shape of the form's unique document instance id, to maintain transaction-specific state information. However, a transaction may also originate through a non-form page such as a printed catalog page, implying, for example, the existence of a virtual "shopping trolley". In this case the relevant transaction state information is tied, indirectly, to the unique id of the user.

The Netpage Registration Server maintains an anonymous relationship between a user and a transaction handler via a uniquely numbered transaction, as illustrated in Figure 4. Whenever the user activates a hyperlink tagged with the "transaction" attribute, the Netpage Printer asks the Netpage Registration Server to translate the associated handler id, together with the pen id, into a transaction id. The transaction id is then submitted to the hyperlink transaction handler. For efficiency, the printer caches transaction ids.

The transaction handler maintains state information indexed by transaction id. It is able to retrieve user-specific state information without explicit knowledge of the user.

Figure 4. User transaction

APPLICATIONS AND BUSINESS MODELS

9 Applications

The Netpage Network has the potential to subsume a wide variety of applications in both traditional and electronic media. This section describes the following possibilities:

- personalized subscriptions to newspapers, magazines and comics
- subscriptions to freelance columns and bulletin boards
- on-demand newspapers, magazines and comics
- on-demand flyers and product brochures
- on-demand books
- television infotainment printing
- e-commerce purchasing from online and traditional catalogs
- delivery of statement and invoices, with online payment
- delivery of secure document (tickets, coupons and licenses)
- perfect copying with copyright micro-payments
- mail replacement
- delivery of greeting cards
- form printing, fill-in, and submission
- delivery of e-mail and facsimile
- on-demand document delivery on corporate intranets
- provision of government services
- Web browsing, searching and printing
- photo album creation
- persistent searchable note taking
- computer printing

9.1 PERSONALIZED SUBSCRIPTIONS

The strength of the Netpage Network lies in automatically delivering subscribed periodicals, at a print quality equalling or exceeding that of their traditional counterparts, with editorial content personalized to individual interests, advertising content localized and targeted to individual localities and demographics, and advertising directly linked to detailed product information and product purchasing.

9.1.1 Newspapers and Magazines

The Netpage Network offers a new delivery mechanism to the $24 billion newspaper and news magazine market which is both more cost-effective than centralized printing and distribution, and allows more fine-grained targeting of advertising.

The simplest form of news personalization involves selecting a news publication and choosing which daily and weekly sections to receive. The simplest form of advertising personalization is tuned to the demographics associated with the subscriber's ZIP or ZIP+4 code. Even with these entry-level forms of personalization, the Netpage Network offers a compelling distribution model for news.

Users receive paper publications with the usability of their traditional counterparts, but with interactive properties. At the press of an on-page button, a user can print an article giving the background to a news story, print a personalized product brochure, or add a product to the virtual shopping basket.

Any magazine normally printed on lightweight paper stock is equally well-suited to distribution on the Netpage Network. However, since Netpage Printers don't carry heavier paper stocks and don't provide wrap-around binding, the Netpage Network is less well-suited to the distribution of so-called "glossy" magazines.

9.1.2 Freelance Columns and Bulletin Boards

Users can choose to subscribe to individual freelance columns, cartoons, etc. These can be integrated into a user's daily news document, or printed individually. Freelancers can choose to receive micro-payments from their subscribers, freeing them from maintaining their own subscriber databases. The Netpage Network provides mechanisms for handling micro-payments.

Users can also subscribe to the "bulletin boards" of friends: collections of news, announcements, pictures etc., which work much like freelance columns.

9.2 On-Demand Publications

The Netpage Network can deliver, on demand, current and back issues of periodicals normally delivered on subscription, including newspapers, magazines, and comics. To maintain the interactivity of all Netpages ever printed, the Netpage Network keeps all published content online at all times. Unlike the Web, where hyperlinks become unreliable over time, content on the Netpage Network never expires.

9.2.1 Flyers and Product Brochures

The Netpage Network makes high-quality flyers and product brochures instantly available, linked to advertisements and entries in printed catalogs.

Brochures are always up-to-date, and link to e-commerce, e-mail, and automatic telephone call-back. Brochure links can provide "click-through" fees to linking documents, and subsidized printing to users.

9.2.2 Books

Users can obtain the latest best-sellers or rare "out-of-print" (a soon-to-obsoleted term) editions on demand, printed in column format with a text size chosen by the user. A typical 300-page paperback fits on as little as 40 sheets of Letter paper. Slip-on covers are available for robust handling.

Titles which have outlived their copyright period are available for free. Other titles are heavily discounted for Netpage delivery, since publishers avoid the costs of printing, inventory storage, and delivery.

Colorful children's books reproduce immaculately. When they've been loved to death, they can be printed again, and again.

Children's coloring-in books and puzzles are available just when they're needed on a rainy day.

9.2.3 Television Infotainment

Users watching infotainment programs on television can print the associated informational material on their Netpage Printer by pressing the "print" button on their remote con-

trol at the appropriate time. The material may be a report on a new medical procedure, the plans for a do-it-yourself bookshelf, or a list of top investment opportunities. It may also be a subscription form for the publication the program is promoting.

The television and/or remote control is suitably modified or augmented to route the print request to the Netpage Printer.

9.3 E-COMMERCE

9.3.1 Online Purchases

The Netpage Network supports a similar level of online purchasing as the Web, but in paper-based medium which presents like a high-quality printed catalog.

A user can navigate the retailer's online Netpage catalog, printing catalog pages as they're needed and adding items to a virtual shopping trolley. The contents of the trolley can be listed at any time, and items can be struck from the list at the stroke of the pen. Pressing a "proceed to checkout" button at any time elicits a completed order form just waiting for the user's signature. The payment card account number is securely shown in the usual 1234 56** **** *789 format. The user's signature authorizes the payment.

9.3.2 Catalog Purchases

Rather than buying from an online Netpage catalog, the user can select items from a traditionally-printed catalog which contains active Netpage links.

9.3.3 Statements and Invoices

Statements and invoices can be securely and auditably delivered, and can be automatically filled in with the user's default payment details without the sender knowing those details.

The user's signature can authorize the payment as normal.

9.3.4 Secure Documents

Retailers can securely issue tickets and coupons over the Netpage Network, printed with difficult-to-forge watermarks.

Agencies of various kinds can issue licenses printed with watermarks and the user's own photograph.

As described earlier, recipients can verify the authenticity of secure documents using a standard Netpage Printer or a special verification pen.

9.3.5 Copyright Copying

Any printed version of a Netpage document becomes an easy means to printing another perfect copy. When a copy is made, the Netpage Network can automatically transfer a micro-payment from the copier to the copyright holder.

Trivial copyright fees are universally respected but seldom paid because of the inconvenience. The Netpage Network offers micro-payment convenience and the quality of an original copy.

9.4 COMMUNICATION

9.4.1 Mail

The Netpage Network, once widely subscribed, can be used to deliver numerous instances of regular mail-outs, particularly statements and invoices as discussed in Section 9.3.3. Once again, the Netpage System only delivers mail from authorized sources.

The United States Postal Service delivers 107,000,000,000 pieces of First Class mail each year [86], a large number of which are regular in nature.

9.4.2 Greeting Cards

A user can select a greeting card from an online catalog, add a handwritten message, and dispatch it via the Netpage Network. Cards can be addressed to other Netpage users, and to normal postal addresses. In the latter case the card is printed at the service center closest to the recipient, automatically placed in an envelope, and mailed through the local mail system.

Netpage users can choose to receive cards from anyone, or only from authorized friends.

9.4.3 Forms

Forms of all kinds can be printed on the Netpage Printer, filled in by hand, and submitted directly over the Netpage Network. Submission is secure and cannot be repudiated.

Handwriting is automatically recognized by the system. The digital ink of the handwriting is attached to the form in case a human clerk needs to re-interpret the handwriting. Automatic "handwriting bots" on the network can assist with the recognition task, automatically giving the user semi-intelligent feedback to elicit disambiguation.

Any interactive Netpage "application", including e-commerce and e-mail, uses forms of various kinds.

9.4.4 E-Mail

E-mail forms can be printed on demand and filled in by hand. The handwritten address is converted to facilitate delivery, but the rest of the message is delivered as digital ink, just as the user intended. If the recipient is computer-based rather than Netpage-based, all of the handwriting can be automatically converted, with the digital ink sent as an attachment, since it may contain hand-drawn diagrams etc.

Netpage users can choose to receive e-mail from anyone, or only from authorized friends.

9.4.5 Facsimile

Facsimile forms can be printed on demand and filled in by hand. The handwritten telephone number is converted to facilitate delivery, but the rest of the message is delivered as bitmapped digital ink, just as the user intended.

9.5 CORPORATE AND GOVERNMENT

9.5.1 Corporate Intranets

An organization can use a private intranet-based Netpage Network to implement a document repository and efficiently distribute documents on demand.

9.5.2 Government Services

Government can provide access to services via the Netpage Network. The network can obviate the need to visit government offices to obtain forms and submit forms, and the network can be used to efficiently deliver the results of submissions.

9.6 PERSONAL

9.6.1 Web Browsing, Searching and Printing

Users can browse the World Wide Web via their Netpage Printer using paper and pen as the user interface. Netpage forms can provide emulation of HTML forms. Only dynamic media objects may fail to print meaningfully.

A Netpage Printer can be the ideal output device for documents encountered while browsing the Web, whether the browsing is terminal-based or Netpage-based. An increasing number of print-ready documents are being published on the Web.

Entire Web sites can be compactly formatted for print, since a printed page has a much greater information-bearing capacity than a computer screen, and perused in a more leisurely fashion. The Netpage Formatting Server automatically adds a table of contents and an index of referenced pags.

Links on Web pages remain active when printed on Netpages. Clicking on a link causes the linked Web page to be printed.

Web pages and search results are printed on both sides of the paper, allowing one sheet of paper to hold the equivalent of ten SVGA screens of information. Thus the user sees 200 links on paper, compared with only 20 on a computer screen.

9.6.2 Photo Album

A Netpage Printer can be enhanced with an infrared data connection (IrDA) to allow it to accept images wirelessly from a digital camera. Alternatively, digital cameras with USB ports can be directly connected. Images can be automatically archived on the Netpage Network, and individual photos, both regular and poster-sized, can be printed at photo quality on the printer.

Smart layout software can assist with the interactive creation of photo album pages, ready for insertion in a cumulative family album.

9.6.3 Notebook

A pre-printed Netpage notebook can be used to capture handwritten notes in a persistent fashion. Notes are captured on the network and are optionally timestamped and signed by a certificate authority to provide undeniable proof of priority, for example with respect to

a patent application. Since the handwriting is recognized and converted to text for indexing purposes, the notes can also be searched by keyword, date, etc.

9.6.4 Examination Papers

Students can sit exams in a controlled manner while physically distant from the classroom.

An examination paper can be delivered at a specific time to a distributed collection of students. The students can then be required to answer the questions on the paper within the alloted time. Their answers can be automatically captured at the end of the allotted time using the "blackboard" form model, and in many cases automatically graded. Once again, unrecognized input can be routed to a human operator.

9.6.5 Computer Printing

A Netpage Printer can be the ideal output device for a personal computer or workstation, whether directly-connected or on a local-area network. The speed of the Netpage Printer and the quality of its output make it attractive to existing computer users.

The Netpage Printers have a USB port as a standard feature.

10 Business Models

10.1 System Principles

The Netpage Network leverages the open technology and extensive infrastructure of the Internet. The widespread acceptance and growth of the Netpage Network is predicated on open competition rather than monopolistic practices.

However, to provide an incentive to early investors, semi-exclusive licenses to Memjet-based Netpage Printer designs will be offered, as well as licenses to manufacture paper and ink consumables.

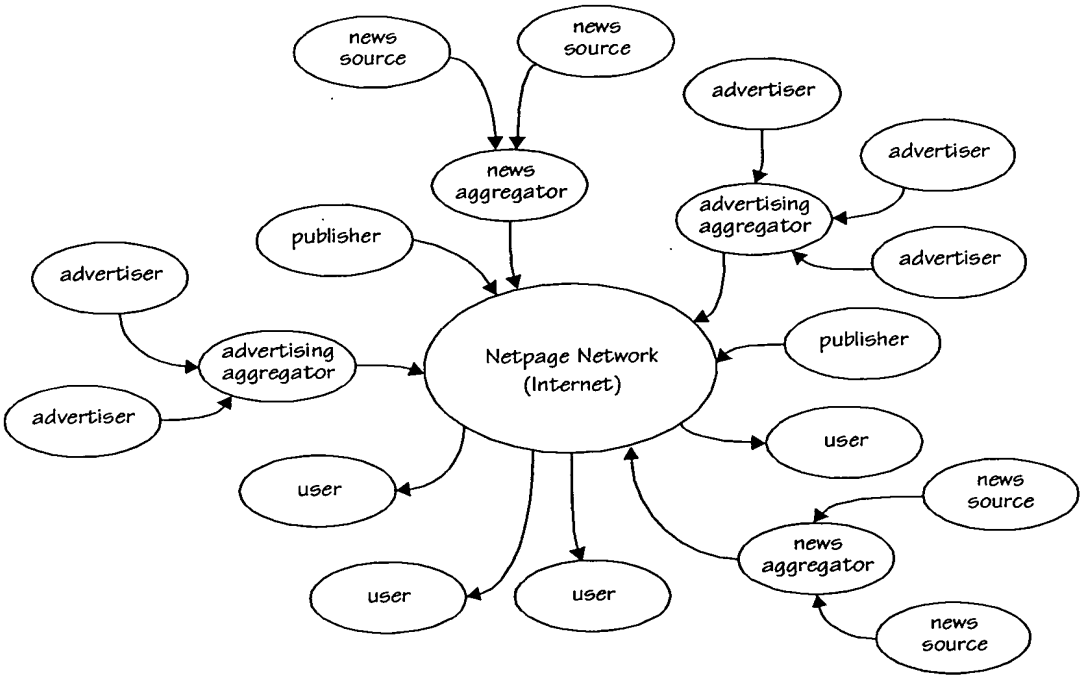


Figure 5. Open structure of Internet-based Netpage Network

The network supports any number of independent participants, some of which have complementary roles, and some of which compete. The open structure of the network is illustrated in Figure 5. Content-related participants include the following:

- news sources
- publishers
- news aggregators
- freelance artists, writers, cartoonists
- direct mailers
- advertisers
- advertising aggregators
- banks
- merchants

Infrastructure-related participants include the following:

- server suppliers
- network storage providers
- communications carriers
- Internet service providers (ISPs)
- printer dealers
- printer installation and servicing companies
- ink and paper consumables dealers
- consumables delivery companies

Technology-related participants include the following:

- research and development
- chip makers (printheads, controllers, QA)
- printer manufacturers
- ink and paper consumables manufacturers

The strength of the network lies in the fact that publication and delivery are completely decoupled. This allows the delivery infrastructure to grow independently of the participation of publishers.

10.2 BOOTSTRAPPING

Because consumers are unlikely to be motivated to acquire a Netpage Printer until a variety of publications and services are available, and because publishers will wait for an installed base before participating, the key to bootstrapping the network is to bundle the printer with a publication or service subscription, and possibly trimming profit margins in the growth stage.

There are several ways the manufacturing cost of the Netpage Printer, assumed to be well below $100, can be subsidized. Printer-based distribution can eliminate existing distribution costs, offsetting the printer cost. The printer can provide a new mechanism for delivering advertising, with advertising profits offsetting the printer cost. And the printer cost can be built into the subscription fee for a publication or service.

The cost of printing and delivering a newspaper normally exceeds the price of subscription [34,66]. The real profit lies in the advertising. The cost of the Netpage Printer is easily exceeded by one years' cost savings, allowing a Netpage subscription, including a "free" printer, to be priced lower than a traditional subscription. A Netpage subscription offered to a customer already on the network would be priced correspondingly lower still.

If the publication or service delivered via the Netpage Printer is sufficiently lucrative, then the publisher or provider may be able to subsidize not only the printer itself, but also its running costs. This can include Internet access, paper and ink consumables, and servicing. Demographics-informed advertising may fall into this category. The more information customers reveal about themselves, the greater the value of the advertising to the advertisers, and so the greater the level of subsidization that can take place.

Early investors who subsidize the installation of Netpage Printers may be able to recover the investment and turn a profit merely by charging other publishers an access fee to the

printers they "own", perhaps for an interim period after installation, according to a network-wide agreement. They may also be able to earn commissions on click-throughs and e-commerce transactions originating on pages printed on "their" printers.

Similar approaches are already emerging in the general Internet market. In the "FreePC" and related models [49], personal computers are bundled with Internet access, and the whole package is fully or partially subsidized by advertising and e-commerce.

Most content-related participants in the Netpage Network, and even Internet service providers, can benefit from directly investing in Netpage Printer deployment.

10.3 MATURITY

Many of the bundling approaches are likely to remain applicable once the network becomes widely subscribed. It is possible that the bundling of the appliance (i.e. the Netpage Printer) with the service (be it Internet access or a publication subscription) will remain the dominant means of distributing the appliance, as it is in the cellular telephone market, and as an increasing number of companies, including IBM [23], are beginning to believe it should be in the personal computer market.

10.3.1 News Publishers and News Aggregators

News publishers with strong brands are likely to be able leverage those brands on the Netpage Network. They have an incentive to do so quickly to prevent newcomers from filling the vacuum and capturing the attendant advertising revenue.

News publishers also have an incentive to migrate to the Netpage Network because it allows them to offer the more fine-grained targeting that advertisers are increasingly demanding, and which they are increasingly seeking elsewhere.

News publishers who create content rather than simply aggregating other sources have a significant advantage, since they offer both unique content and an editorial voice. Users are more likely to choose a single news publication whose content and editorial orientation they find useful, than specifying to a news aggregator how to glue together a number of disparate news sources. And any sufficiently strong news publication brand is unlikely to make its content available to an aggregator, since the aggregator will be taking a proportion of advertising and e-commerce revenue.

The Netpage Network, like the Web, offers lower barriers to entry than traditional publishing media, and this naturally stimulates greater diversity. However, the geographic independence of the network, coupled with built-in mechanisms for localization of publications, allows international, national and regional news publications to more easily compete in local news markets.

The strength of a traditional local news publication lies partly in its local news content and partly in its local retail advertising and classified advertising content. Aggregation of classified advertising is already happening on the Web, and the Netpage Network will make the same thing possible for local retail advertising. Local news publications are therefore likely to be excluded from the direct capture of local advertising, and may instead transform themselves into news gatherers feeding localized editions of larger publications.

10.3.2 Advertising Aggregators

The Netpage Network promises to be the most effective advertising medium ever conceived. It combines the editorial and print quality of traditional publications with arbitrarily finely targeted advertising, and provides a direct link between advertising, product information, and purchasing.

Because personalization and localization are handled automatically by Netpage Publication Servers, an advertising aggregator can provide arbitrarily broad coverage of both geography and demographics. The subsequent disaggregation is efficient because it is automatic.

This makes it more cost-effective for publishers to deal with advertising aggregators than to directly capture advertising. Even though the advertising aggregator is taking a proportion of advertising revenue, publishers may find the change profit-neutral because of the greater efficiency of aggregation.

Because of the finer targeting supported by the Netpage Network, publishers and advertising aggregators have a larger advertising space to sell, leading to greater profits. The linking between advertising, detailed product information, and purchasing, and the corresponding measurability of consumer behavior, leads to greater profits from click-through fees and e-commerce commissions, benefiting publishers and advertising aggregators alike.

Added revenue from these fees and commissions may even allow users' costs - printer, ink, paper, and Internet access - to be fully subsidized.

NETPAGE PRINTER

11 Printer Product Design

Netpage Printers are intended for use in domestic, commercial, corporate and hospitality environments. They are all based on a simple and reliable straight paper path, passing through a Memjet transfer roller printhead mechanism. In most cases the printed page is glued along one edge and adhered to the previous page to form a final bound document that can be 1 page or 500 pages thick. They all interact with the wireless Netpage Pen.

Netpage Printers come in various forms: wall-mounting, tabletop, portable, and pocket versions.

11.1 WALLPRINTER

A low-cost, wall-mounted, base model with a duplex 8½" Memjet printhead array that accepts a full ream of US Letter paper in a vertical format as shown in Figures 6, 8 and 9. Paper is placed into a hinged top tray down onto a sprung platen and registered under edge guides before being closed. Figure 7 shows the access to the paper and ink cartridge.

A replaceable cartridge containing cyan, magenta, yellow, and infrared inks and glue is also accessible when the tray is open. It connects via a series of self-sealing connectors to hoses that transmit ink and glue to their separate locations. The cartridge consists of a thin wall drawn aluminum casing that accommodates four ink bladders and a single glue bladder into an injection molded connector base. This is a fully recyclable product with a capacity for printing and gluing 3000 pages (1500 sheets). It is protected from forgeries by use of an authentication chip [1,2].

When closed, a release mechanism allows the platen to push the paper against the pick-up roller assembly, where it is fed directly into the duplex Memjet printhead assembly. From there, the sheet passes a momentary action glue wheel with powered spike wheels, where it has glue applied to the vertical edge as it passes through. The glue wheel is capped when not in use and is operated by a powered camshaft.

The printed sheet is fed down to a binding platen that operates with a closed steel wire loop system of pulleys, runners and a powered axle. This provides the necessary speed to push the sheet forward onto the rear of a previous sheet, glue/bind it and return to the home position to accept the next printed sheet in less than 2 seconds. A motorized paper tapper assembly aligns the sheets in a simultaneous operation.

When a document is bound and finished, a powered exit hatch with a tamper sensor opens. Plastic foils work together with the hatch to direct the finished document to the back of the collection tray and feed any further documents into the tray without hitting existing ones. The collection tray is molded in clear polycarbonate and pulls out of its socket under a certain loading. Access for removing documents is provided on three sides.

The printer has a main PCB that accommodates all major circuitry components including external data jacks. A flex PCB runs from the main PCB to the paper tray and has three different color LEDs and a push button. The LEDs indicate "on", "ink out", "paper out", and "error". The push button is a "help" interface that prints out a simple instruction sheet and a compact features directory for the user. The unit is powered by an internal 110V/220V power supply that is connected before wall-mounting.

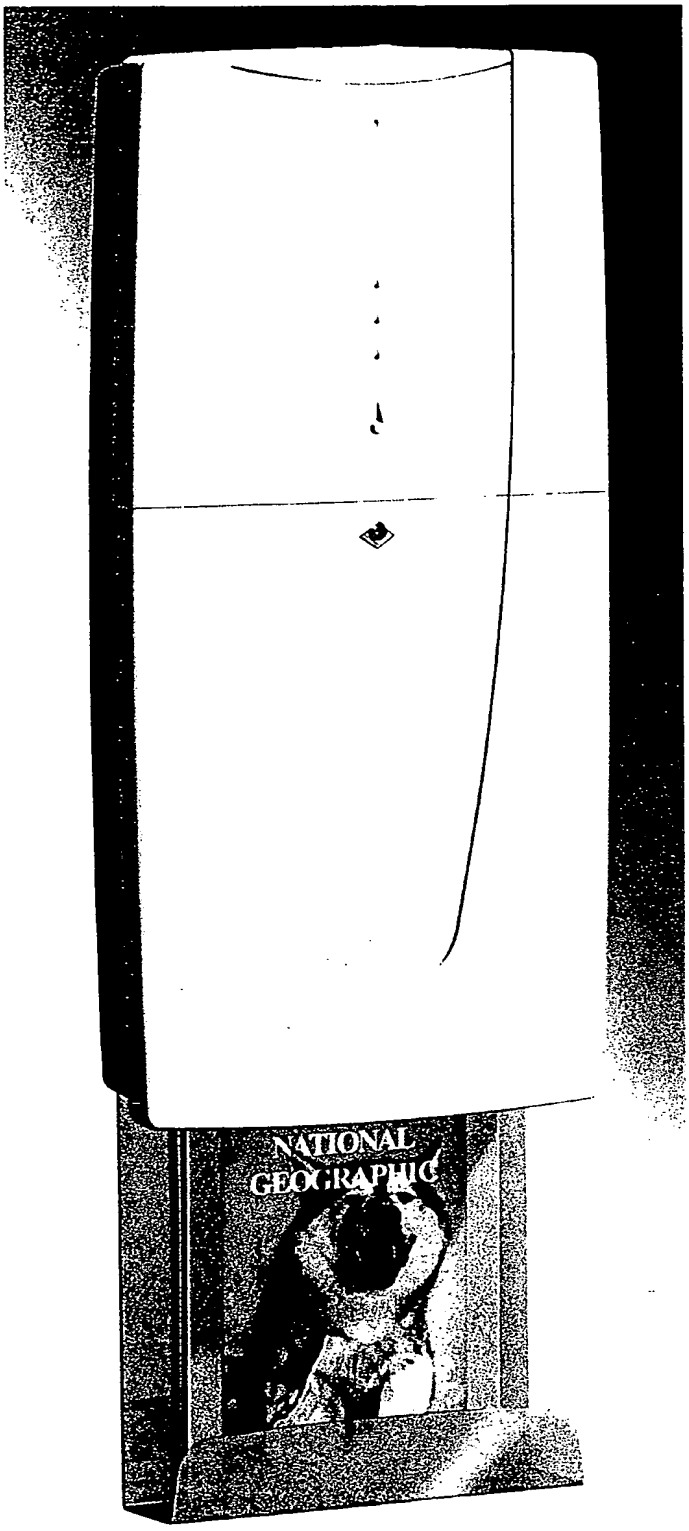


Figure 6. Wallprinter

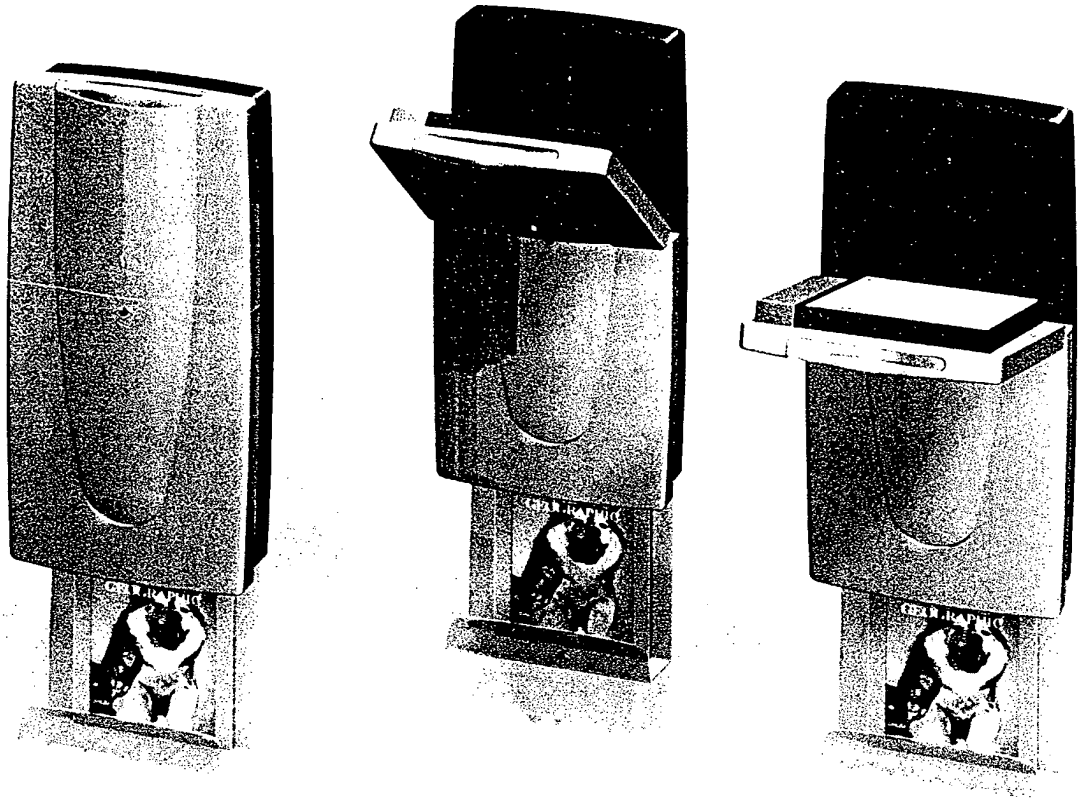


Figure 7. Wallprinter paper and ink cartridge access

The printer has several metal hangers on the rear, which locate into keyhole slots in a metal back plate that is securely fastened to a wall. When the printer has been connected, it is hung onto the back plate and fixed with a locking screw found under the paper tray.

The Netpage Printers are fully customizable in finishes and color as the front moldings clip on to a core chassis and are easily removable.

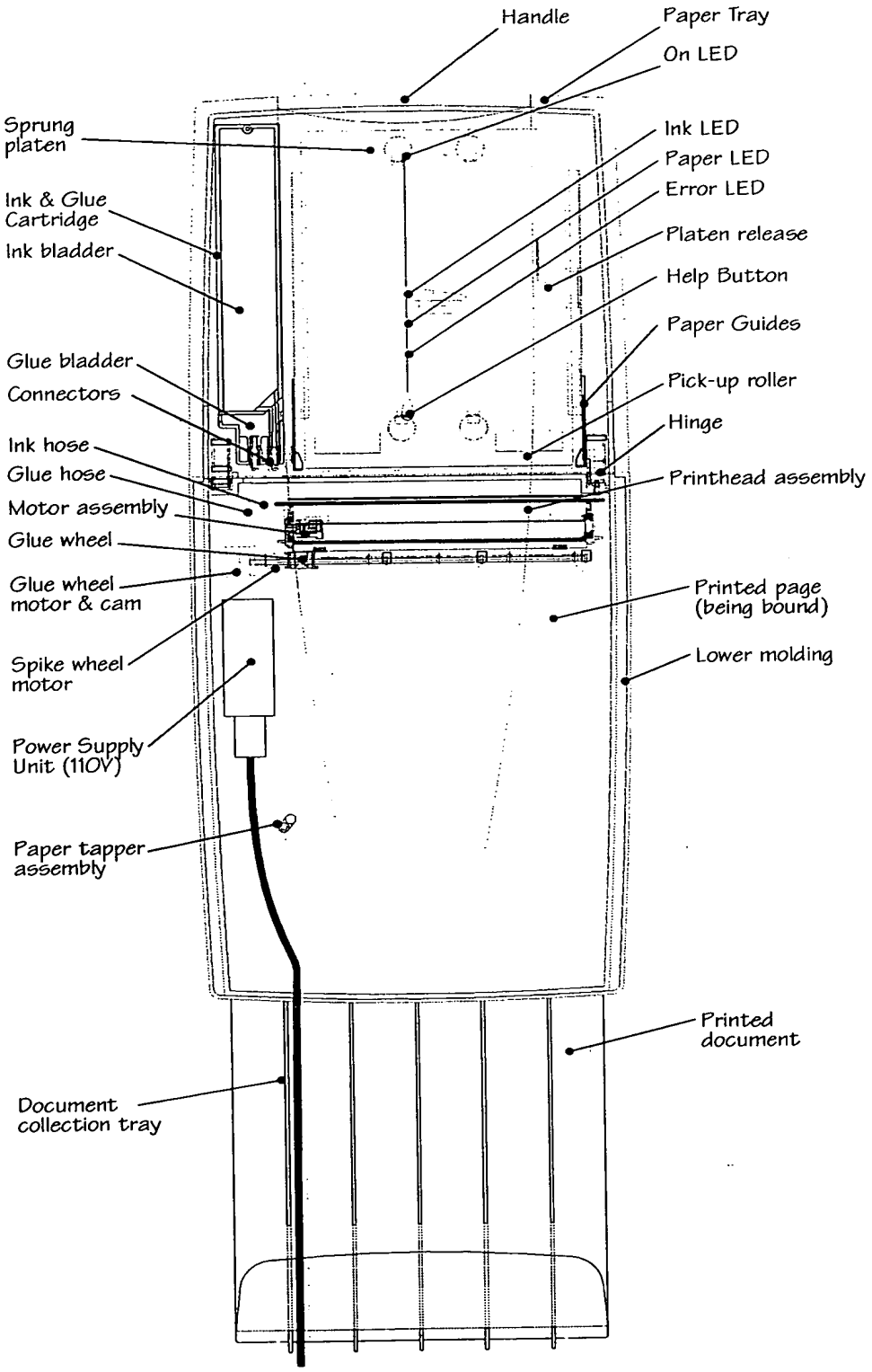


Figure 8. Wallprinter front elevation

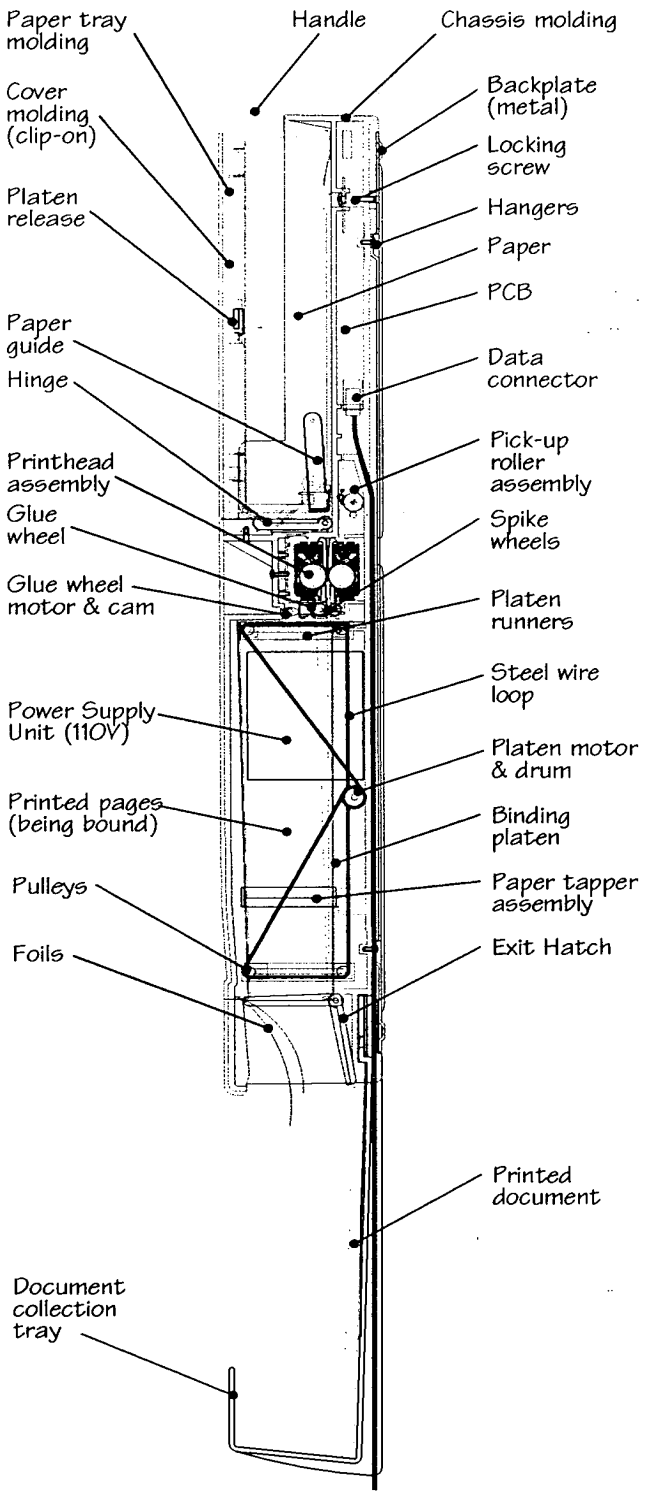


Figure 9. Wallprinter side elevation

11.2 WALLPRINTER PRO

This printer is similar to Wallprinter in most respects, except that it has a duplex 11" printhead assembly, which prints on US Letter paper in a landscape format (see Figures 10, 11 and 12). This means a faster print time and binding time for each page, making for faster overall document delivery.

Another difference is the location of the ink cartridge, which resides above the paper tray rather than down the side. Each page is glued along the horizontal edge by a full-length glue sponge, which is capped when not in use. Operation, printing, and document handling are identical to Wallprinter.

Wallprinter Pro is fully customizable in finishes and color.

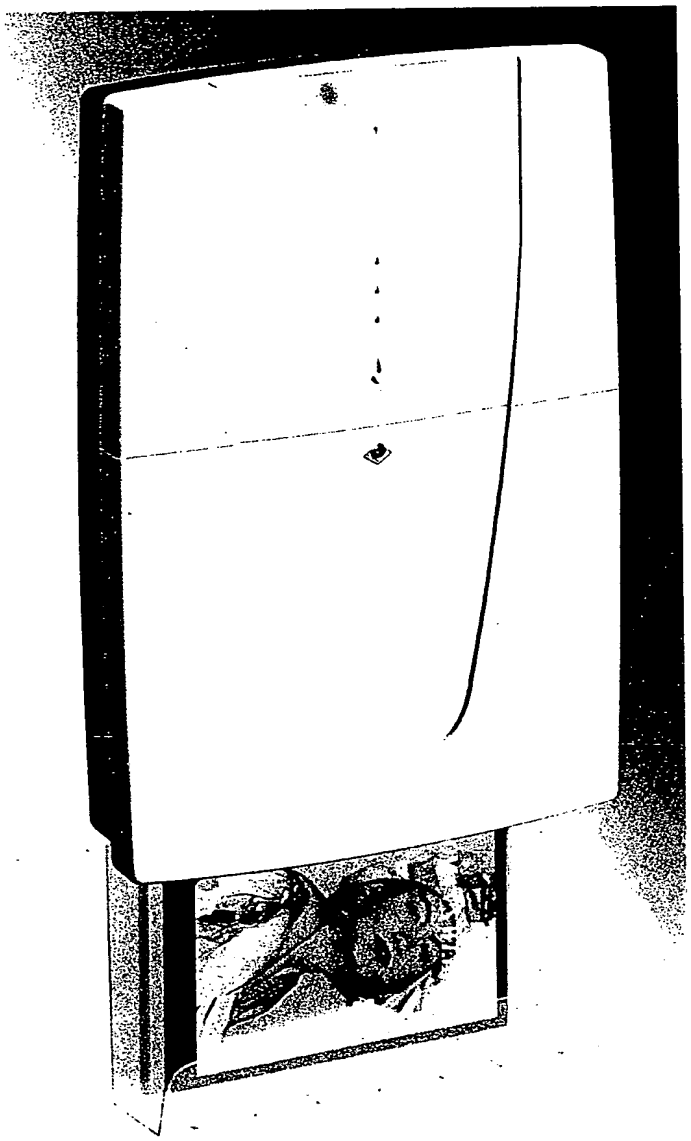


Figure 10. Wallprinter Pro

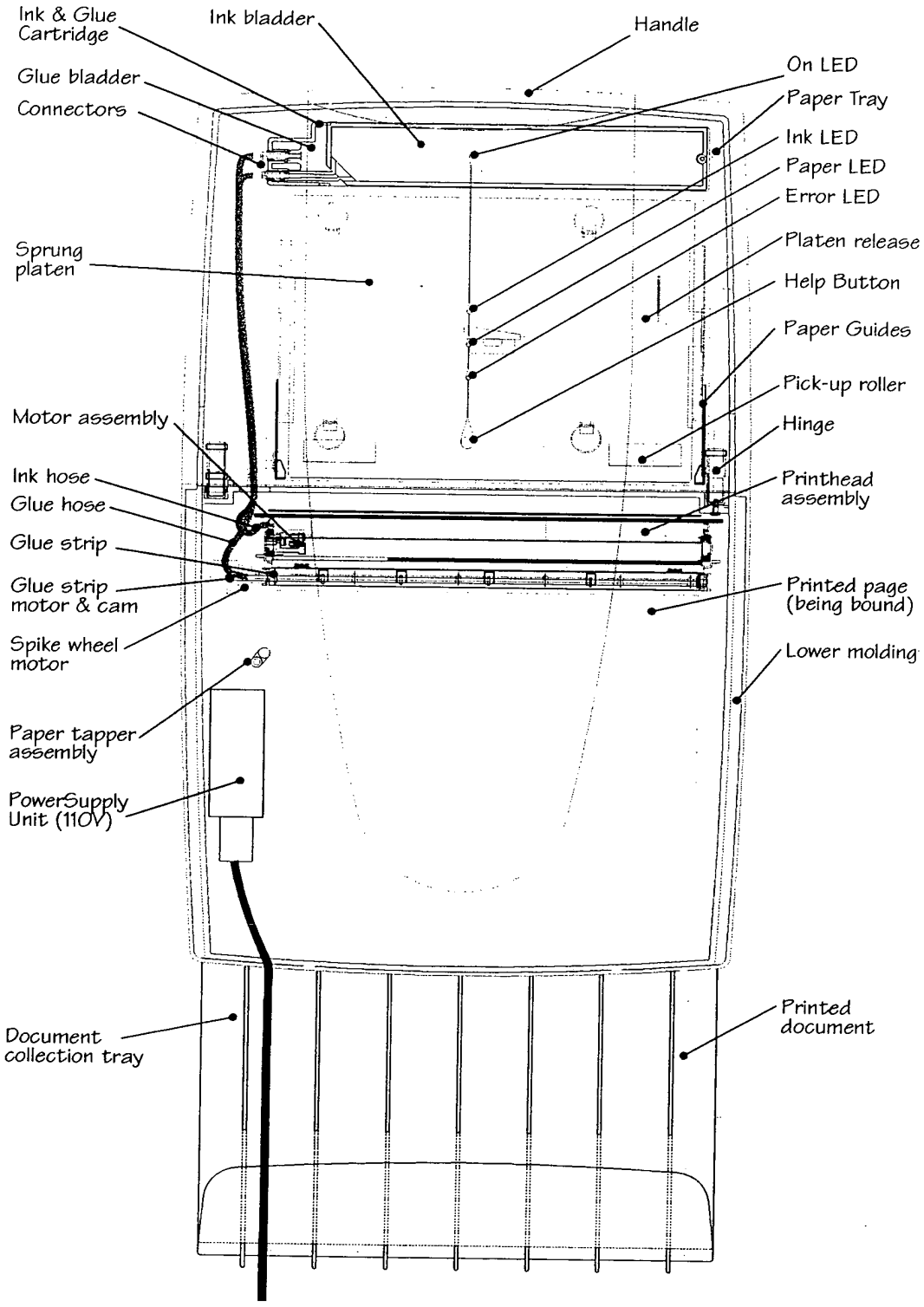


Figure 11. Wallprinter Pro front elevation

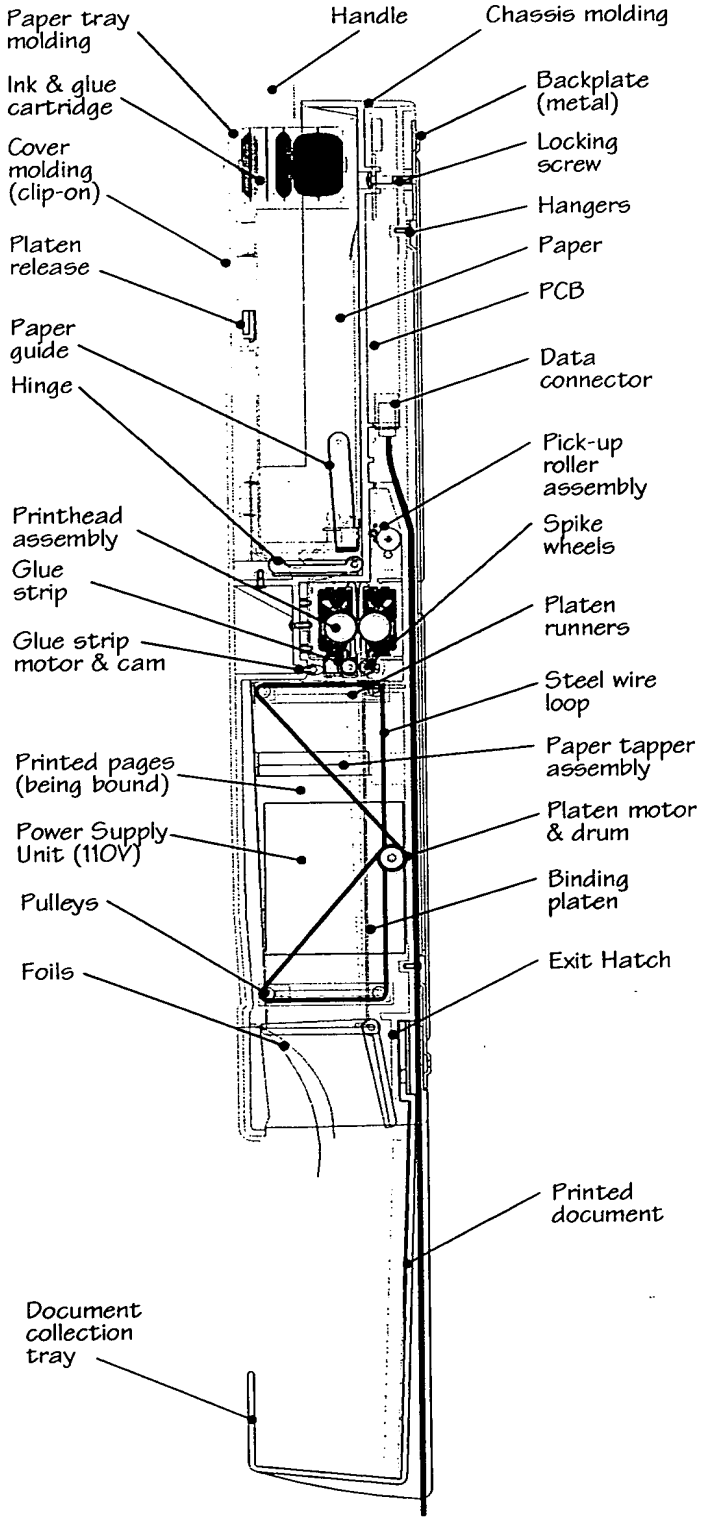


Figure 12. Wallprinter Pro side elevation

11.3 TABLEPRINTER PRO

This printer is a tabletop version of the Wallprinter Pro. Essentially, it is the same printer unit with a base plinth that adds extra functionality, such as USB, parallel port and a power socket (see Figure 13).

Tableprinter Pro is fully customizable in finishes and color.

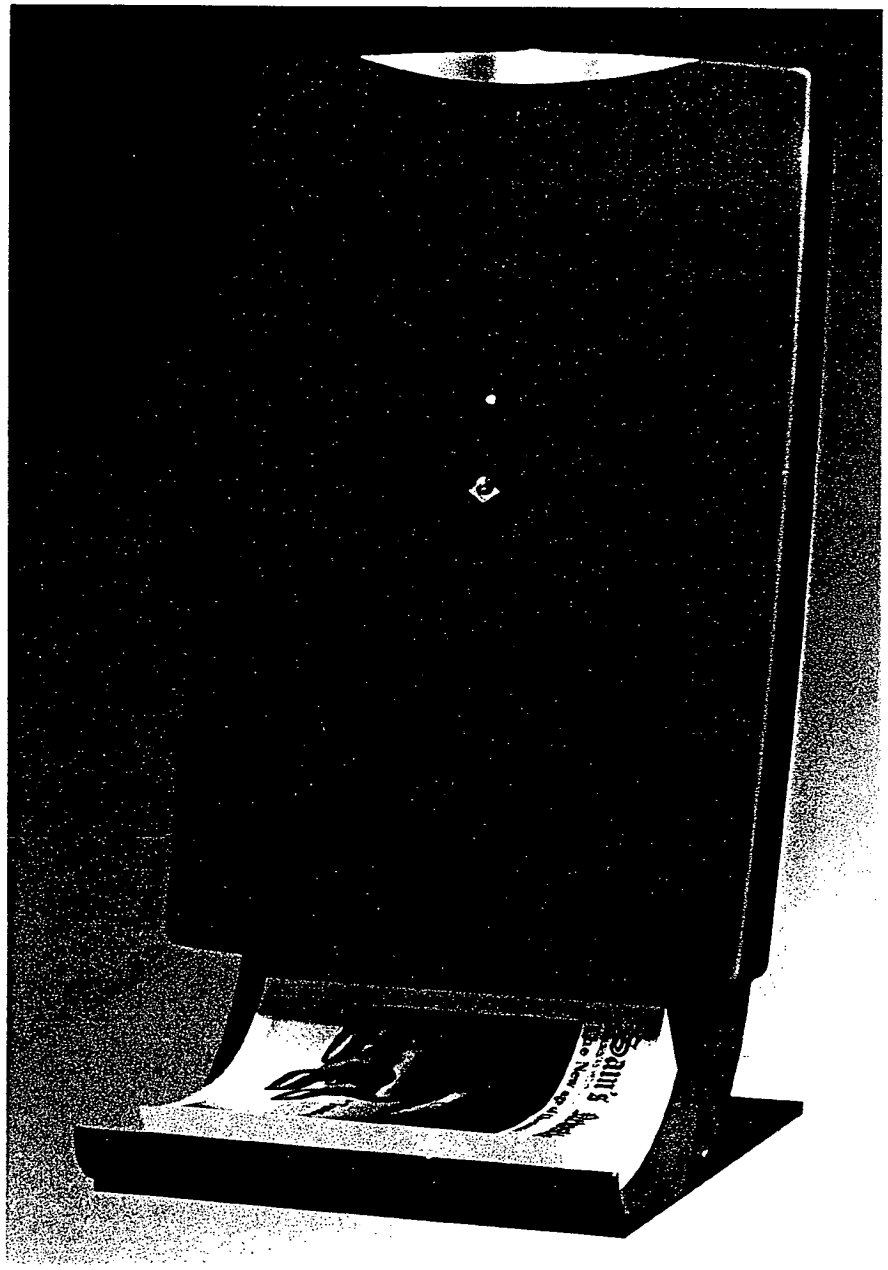


Figure 13. Tableprinter Pro

11.4 WALLPRINTER PRO R

This printer shares the same printing and binding configuration of US Letter landscape format as Wallprinter Pro. The main difference is the media delivery, which is in the form of a large print cartridge cartridge (see Figures 14, 15 and 16). This cartridge accommodates C, M, Y, and infrared inks and glue as well as a 1000 sheet capacity roll of paper. The cartridge can be recharged at nominated outlets when required and it is protected from forgeries by an authentication chip [1,2].

The printer has integral structural metalwork to support its weight and a ball bearing track for easy loading and removal.

Wallprinter Pro R is fully customizable in finishes and color.

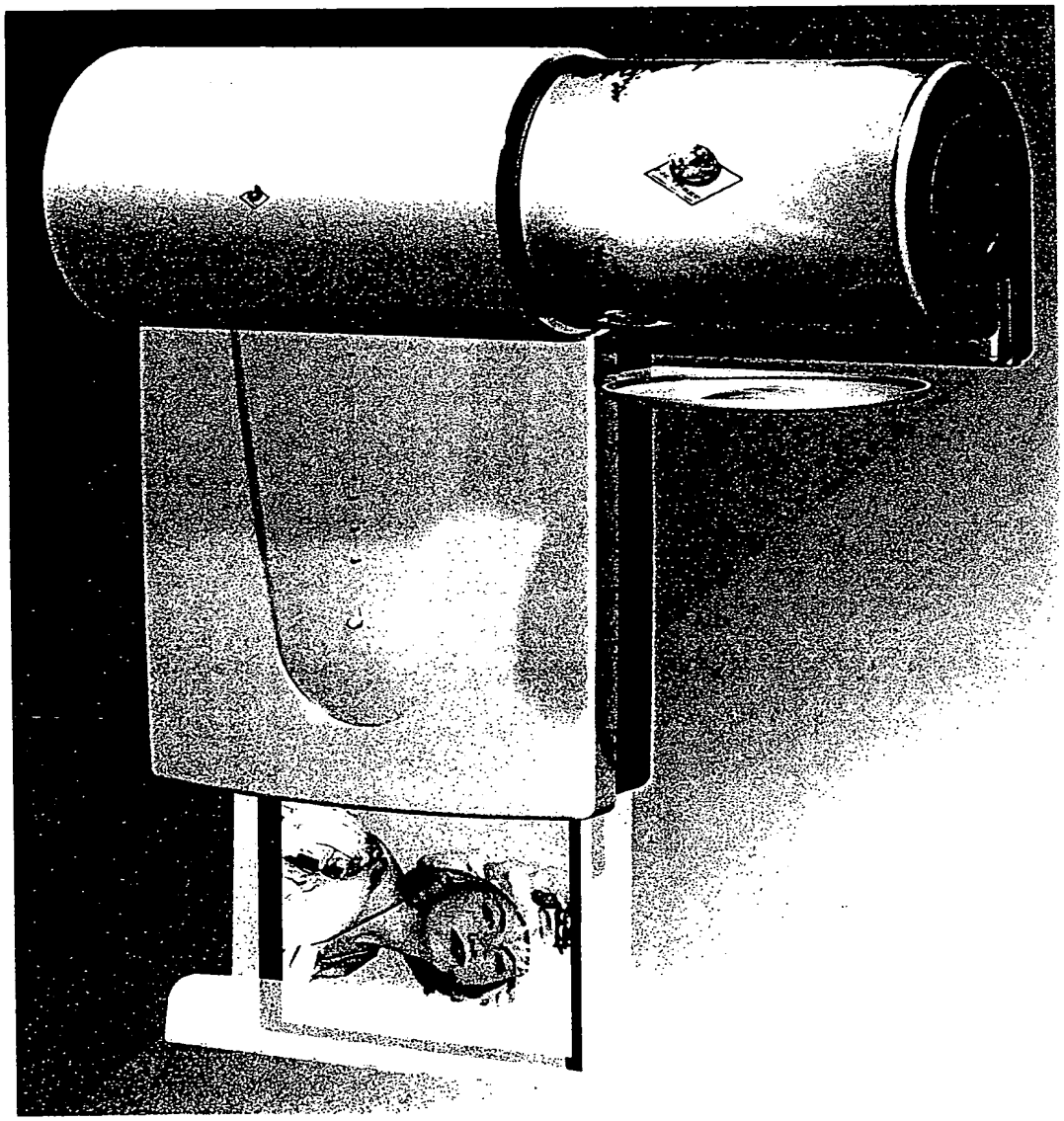


Figure 14. Wallprinter Pro R, with print cartridge extracted

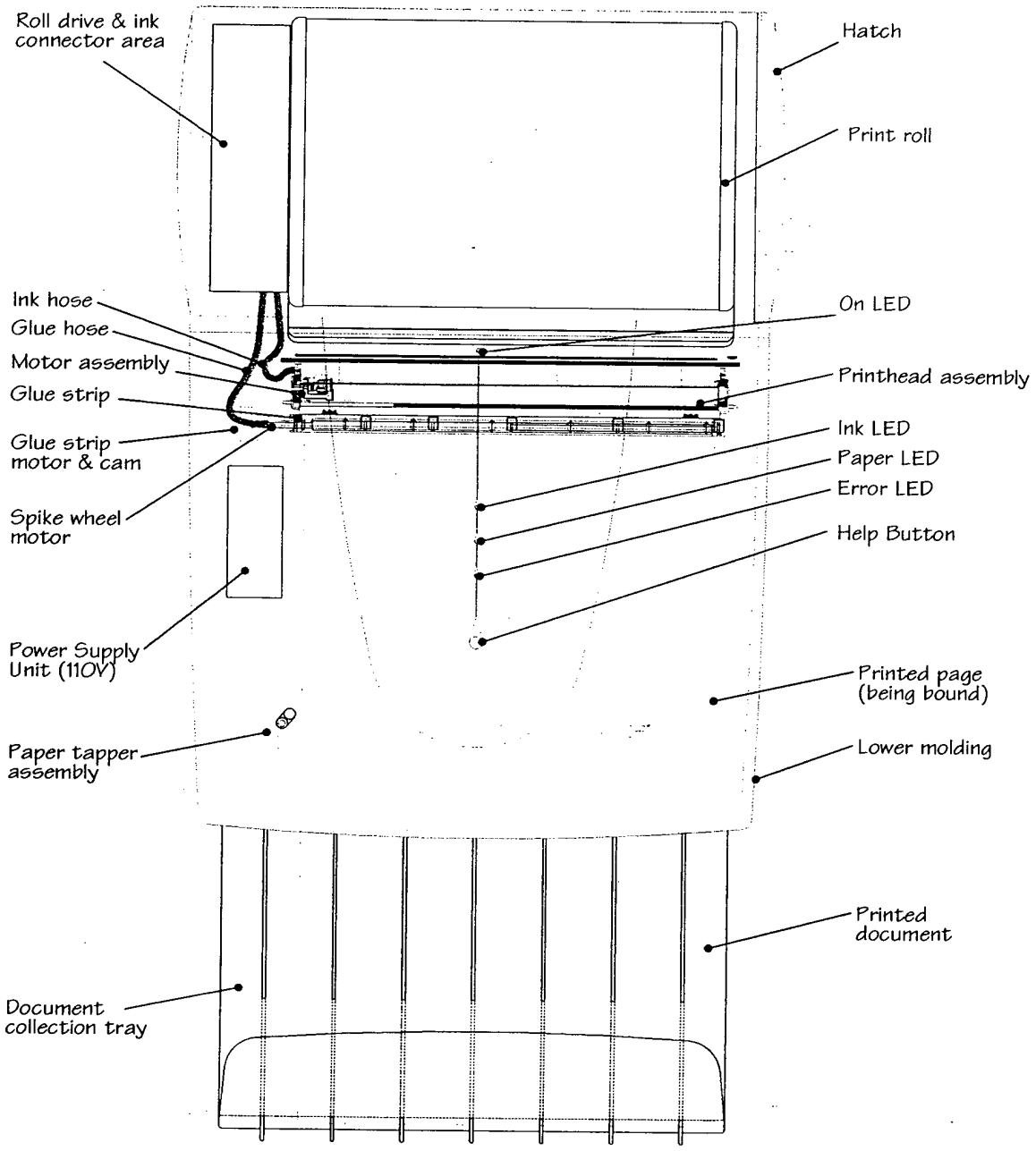


Figure 15. Wallprinter Pro R front elevation

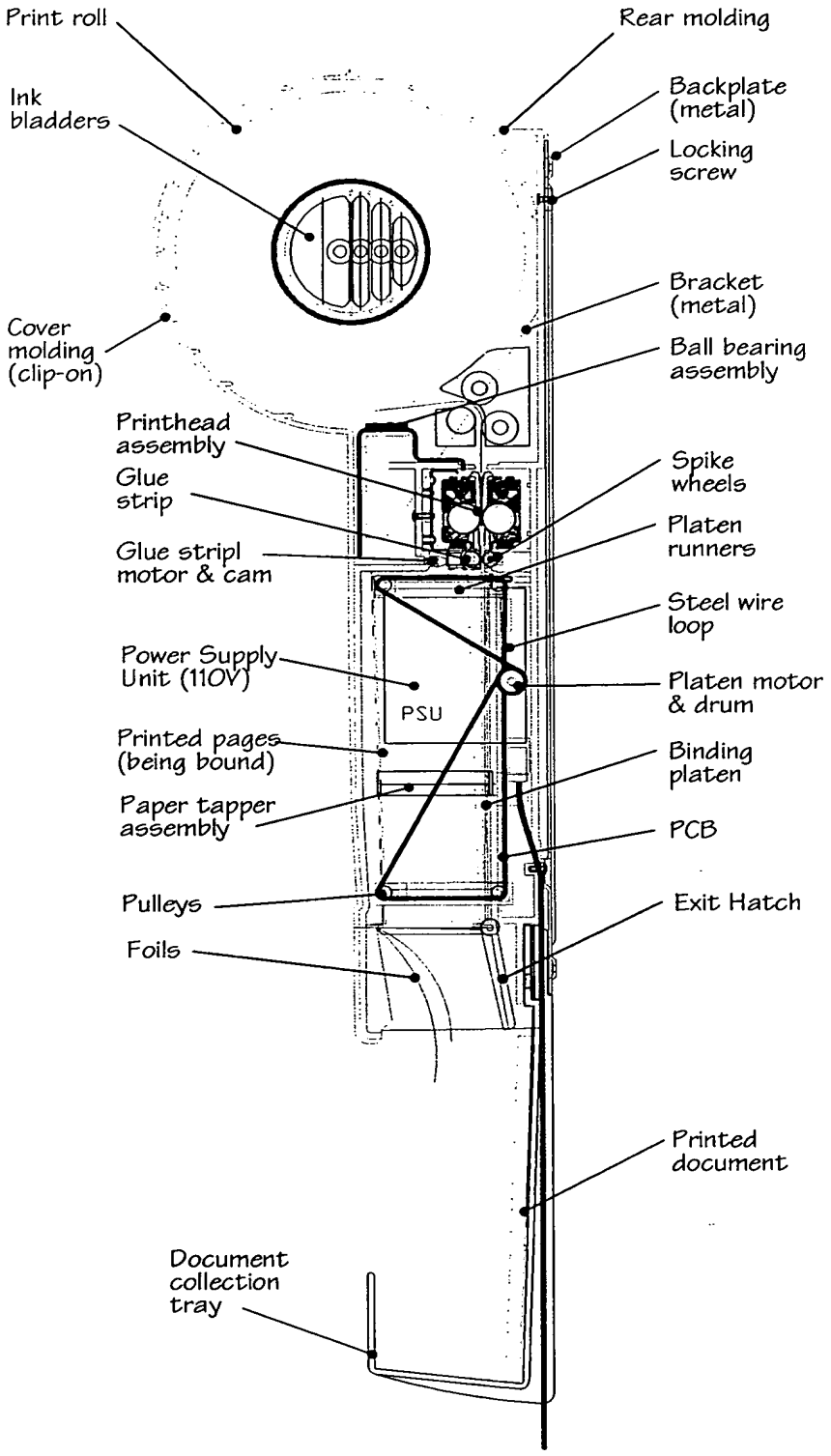


Figure 16. Wallprinter Pro R side elevation

11.5 TABLEPRINTER PRO R

This printer is a tabletop version of the Wallprinter Pro R. Essentially, it is the same printer unit with a base plinth that adds extra functionality, such as USB, parallel port and a power socket (see Figure 17).

Tableprinter Pro R is fully customizable in finishes and color.

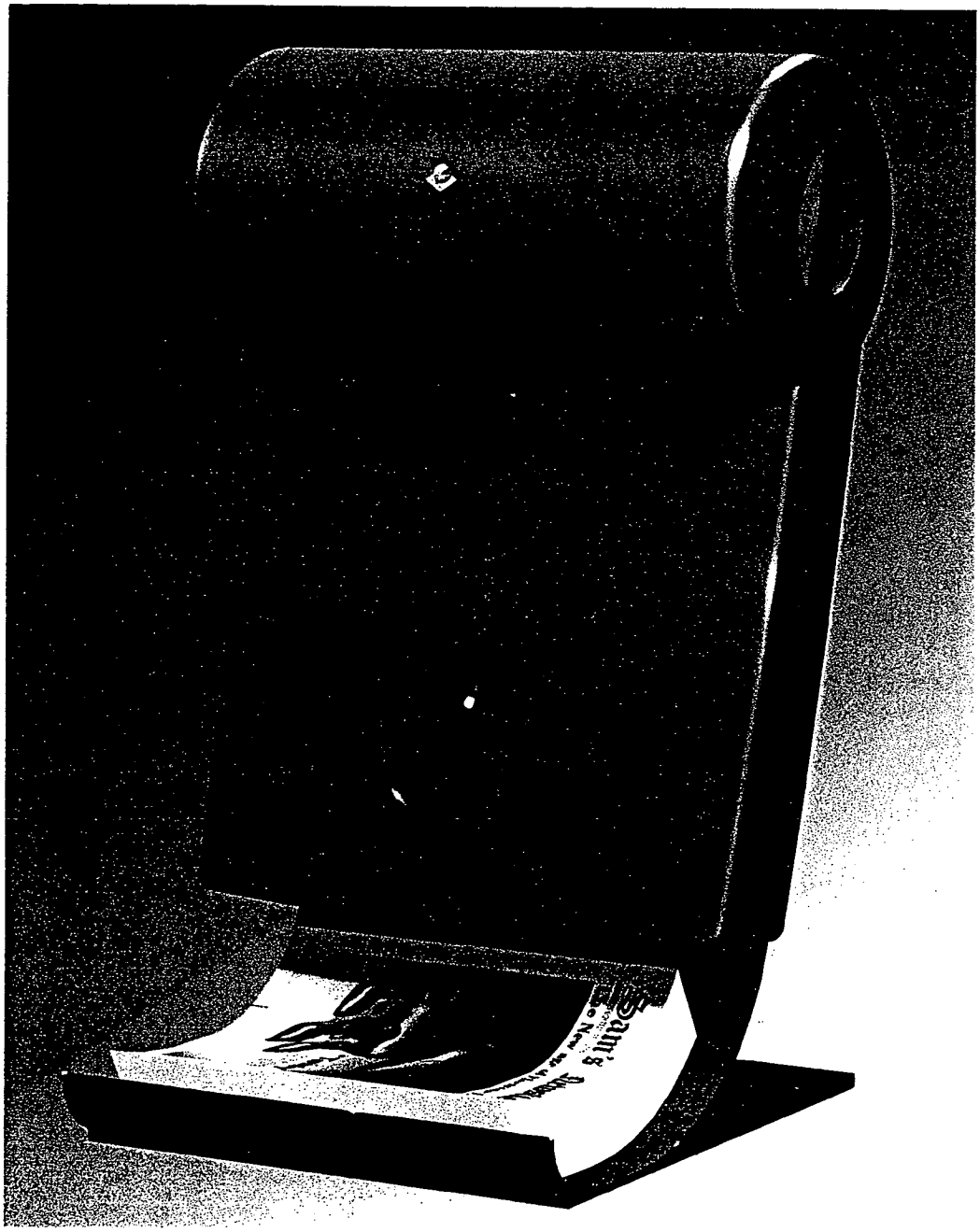


Figure 17. Tableprinter Pro R

11.6 DESKPRINTER PRO R

The Deskprinter Pro R is a vertical format printer based around the large format print cartridge (see Figures 18, 19, 20 and 21). This cartridge accommodates C, M, Y, Infrared inks and glue, as well as a 1000 page roll of paper. The print cartridge is lowered into the top of the unit through a latching hatch. When printing, the paper is fed through the printhead assembly where it is duplex printed and then ejected past a cutter which crops it into sheets. As the sheet passes the cutter, a glue wheel assembly glues along the long edge and powered spike rollers propel the sheet out into the binding area.

The binding area consists of a binding platen which operates with a closed steel wire loop system of pulleys, runners and a powered axle. This provides the necessary speed to push the glued sheet forward onto the rear of a previous sheet, glue/bind it and return to the home position to accept the next printed sheet in less than 2 seconds. A motorized paper tapper assembly aligns the sheets in a simultaneous operation.

When the document has been bound, a series of metal fingers at the top and bottom of the document rotate out of the way and the document is pushed into the out tray area. A powered hatch opens and a motorized ejector assembly pushes the document through the exit slot of the unit. The hatch then closes onto the document ready for removal. Subsequent documents are stacked up in a similar fashion. The user interface is identical to the other printers in the range. A large PCB contains all the processing and external interface components and connects to an internal 110V/220V power supply unit.

Deskprinter Pro R is fully customizable in finishes and color.

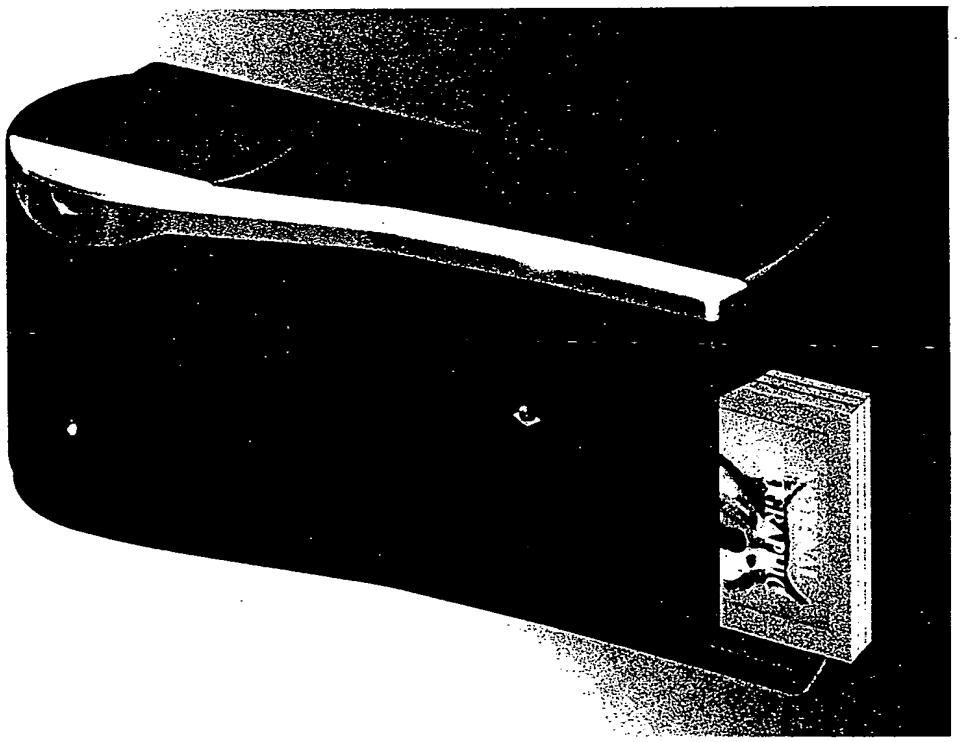


Figure 18. Deskprinter Pro R

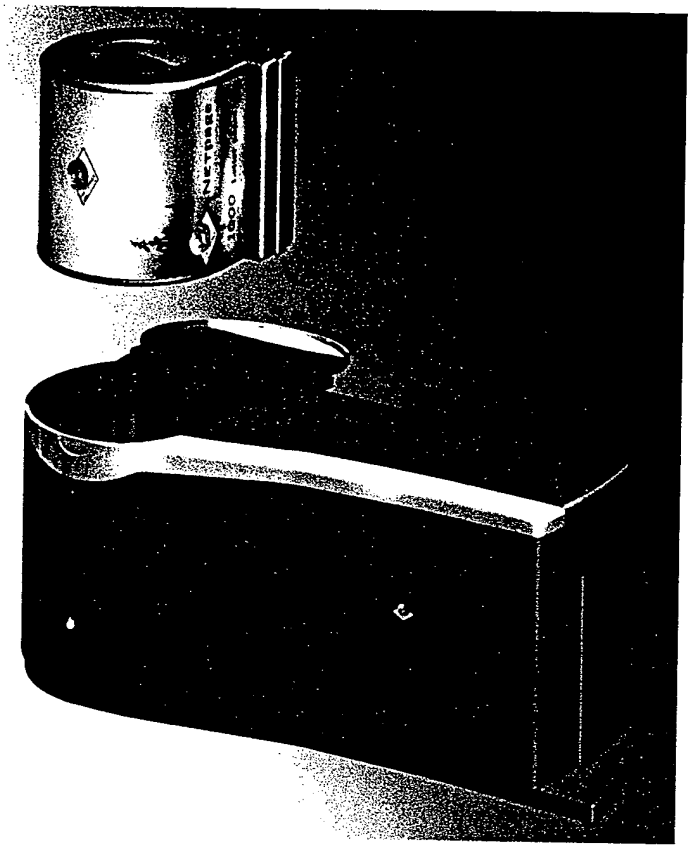


Figure 19. Deskprinter Pro R, with print cartridge extracted

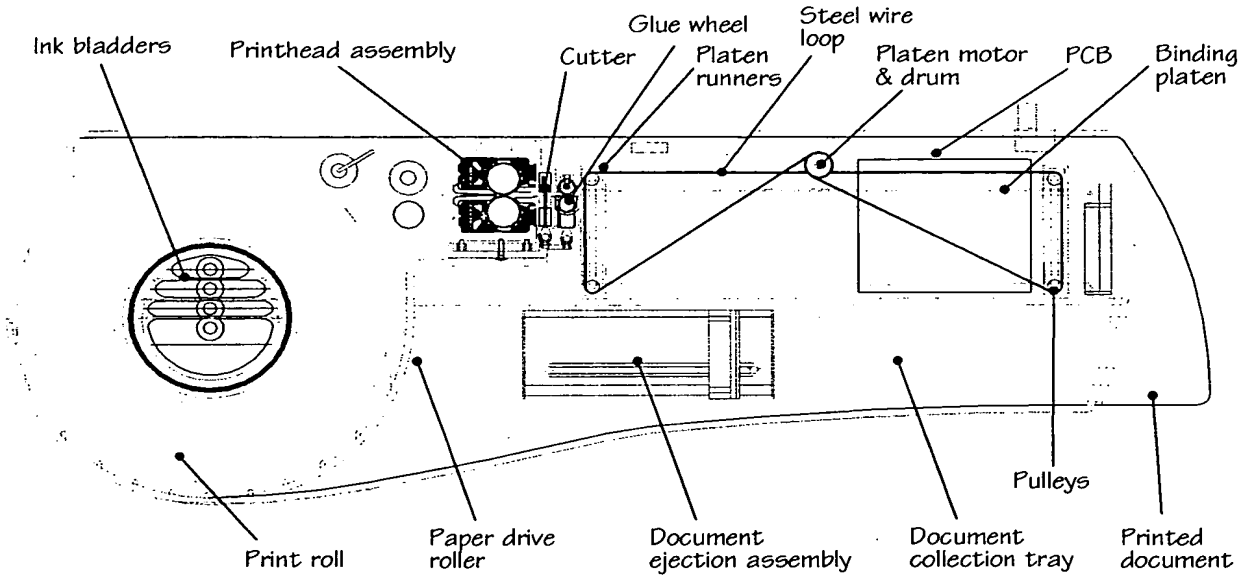


Figure 20. Deskprinter Pro R plan

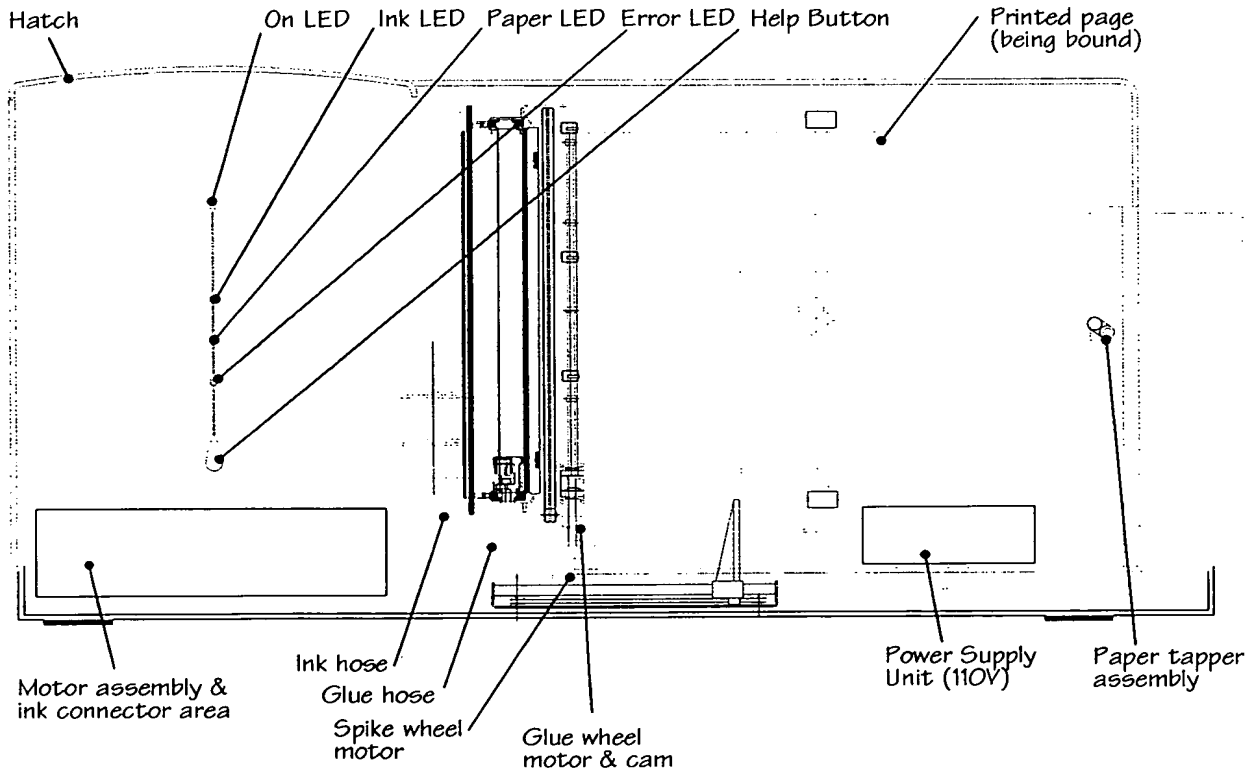


Figure 21. Deskprinter Pro R elevation

11.7 TRAVELPRINTER R

The Travelprinter R is a small, lightweight, versatile and completely portable Netpage Printer (see Figures 22, 24, 25 and 26). It has in-built mobile network communication hardware and software, allowing it to download documents anywhere. Travelprinter also has communications ports for computer interface printing when required.

The printer consists of a front and rear molding with a chassis to accommodate the major components including a lithium battery and an 8½" duplex Memjet printhead assembly.

A compact print cartridge cartridge with C, M, Y and infrared inks and paper is used in the printer, providing 50 US Letter pages or 100 A5 pages. It is protected from forgeries by an authentication chip.

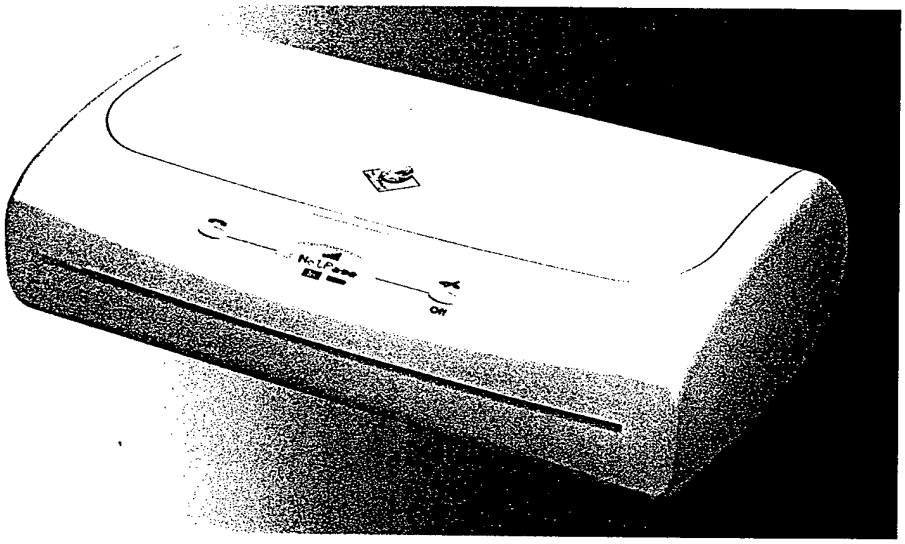


Figure 22. Travelprinter R

A motorized guillotine assembly cuts the media between the cartridge and the printhead and motorized spike wheels eject the finished print out of the unit. A flex PCB runs from the main board to a segment LCD and two push buttons. The LCD shows signal strength, any errors, battery and number of pages left in the cartridge. The buttons allow the printer to either connect to the Netpage Network or to act as a stand-alone printer. A USB interface is provided on the side of the printer along with DC 3V input.

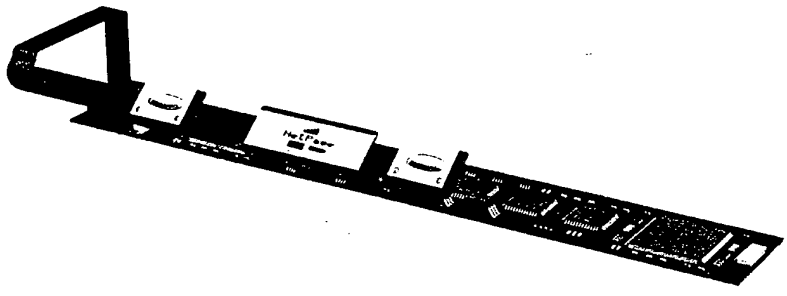


Figure 23. Travelprinter R main and flex PCB, with buttons and segment LCD

Figure 24. Travelprinter R, with print cartridge extracted

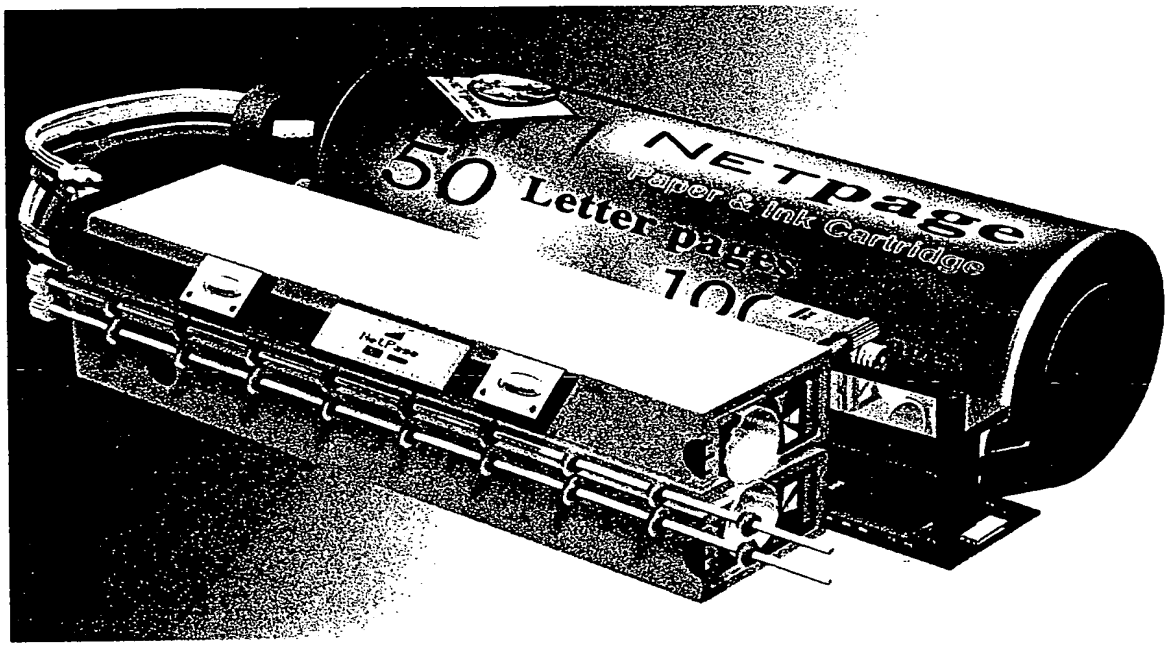


Figure 25. Travelprinter R detail, showing duplexed imaging units and print cartridge

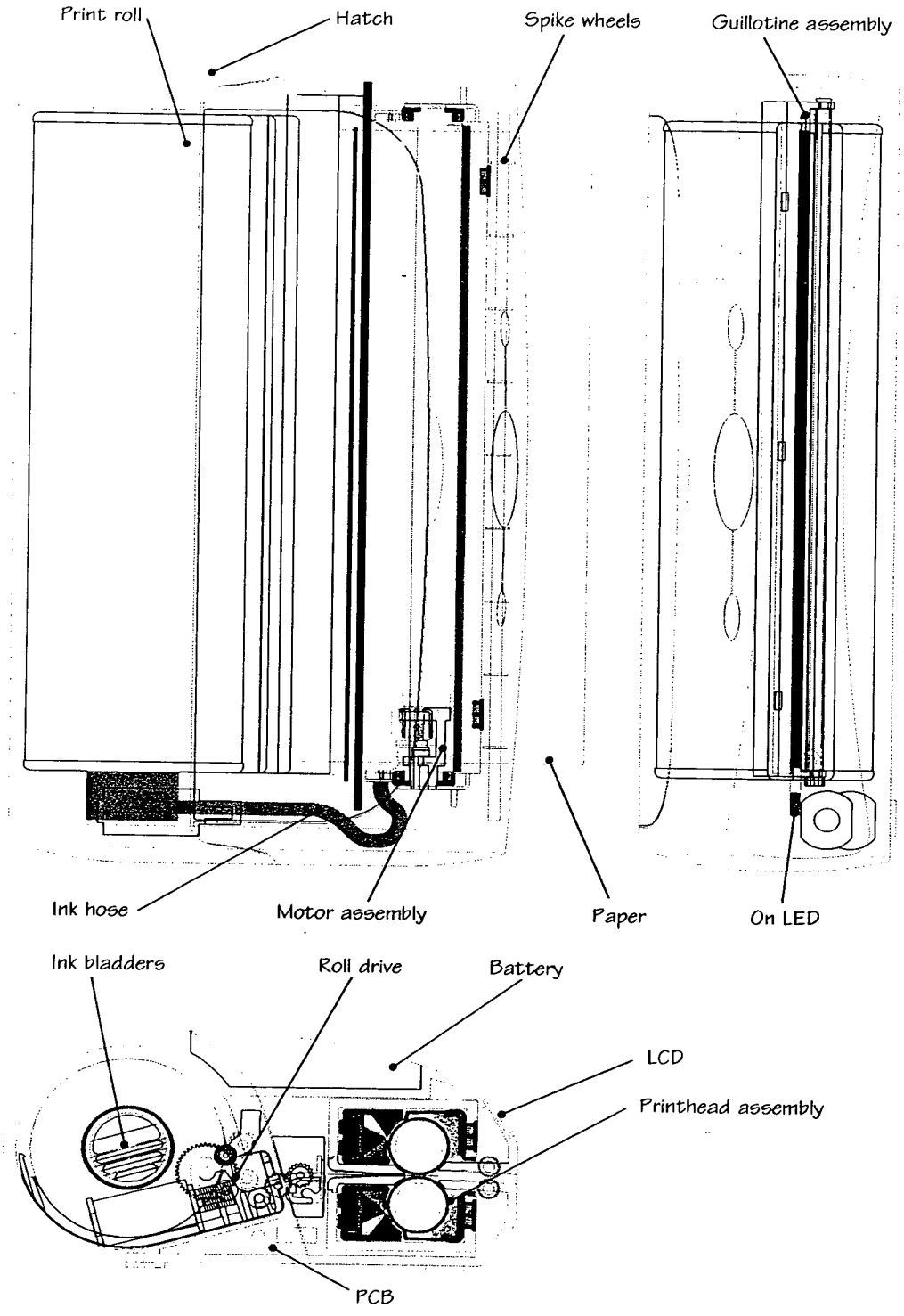


Figure 26. Travelprinter plan and elevations

11.8 MICROPRINTER

The Microprinter is a small, versatile, pocket-size printer/camera (see Figures 27 and 28). It has built-in mobile network communication hardware and software, allowing it to link to the Netpage Network and fetch documents from anywhere. In addition, the product is ergonomically configured and styled as a fully functional digital camera with the standard Classic, HDTV and Panoramic print formats for photography.

The Microprinter accommodates a 4" (100mm) page-width print cartridge with C, M, Y and infrared ink plus 5.4 meters of paper. This means the user can single-sided print 41 Netpages, 36 Classic, 30 Horizontal or 18 Panoramic prints or any combination thereof.

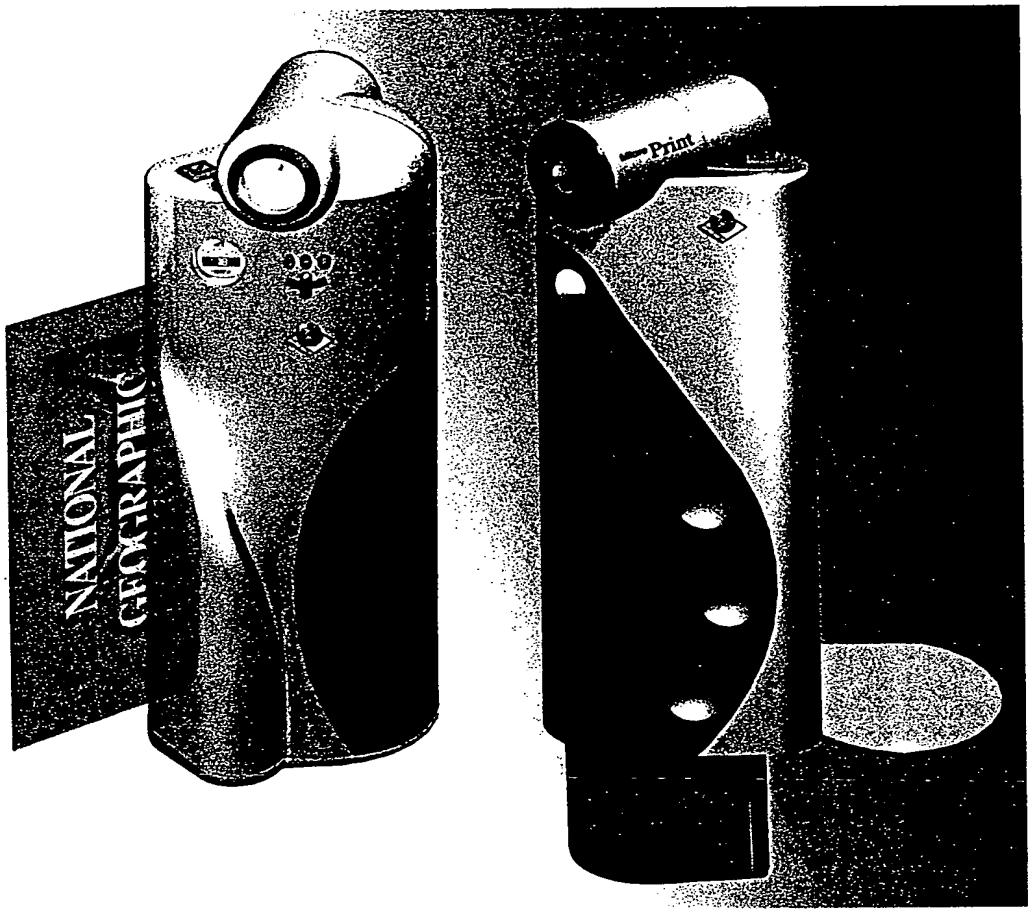


Figure 27. Microprinter rear view with printing in progress (left), and front view with print cartridge extracted (right)

The Microprinter is a fixed-lens camera with auto focus and digital photo enhancement capabilities, which allows the user to take sophisticated photos very simply and easily. A print button allows the user to print and reprint a photo or Netpage.

The Microprinter consists of two main front and rear moldings and a hatch. The main moldings accommodate the lens assembly, viewfinder optics and a rigid PCB. A flex PCB runs from the main board to the imaging chip, an LCD, the printhead assembly, the print-

head capping mechanism, a cutter assembly, the roll motor drive, various control switches and a CR2 battery.

The Microprinter is fully customizable in finishes and color.

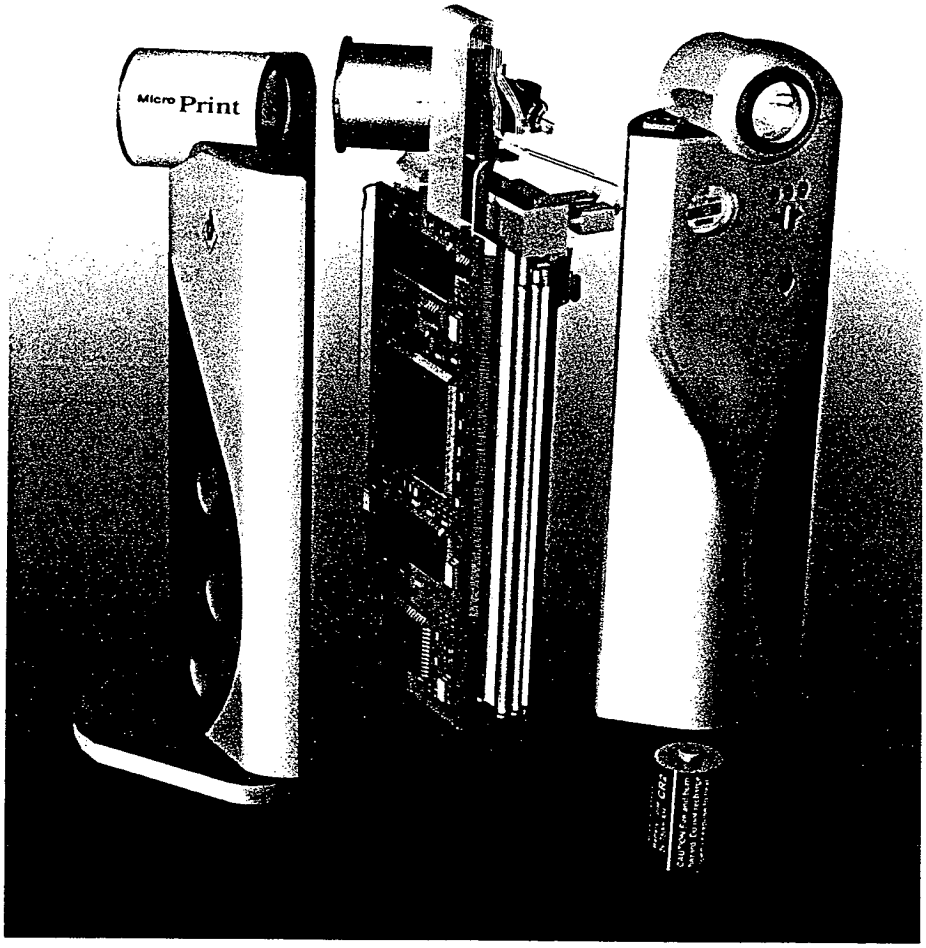


Figure 28. Microprinter exploded view

11.9 NETPAGE PEN

The Netpage Pen is an intelligent, interactive writing implement that communicates with the Netpage Printers and the Netpage Network. It allows the user to write normally with an ink nib or use a non-marking stylus. Rotating the top of the pen selects between them.

The pen has a distinctive shape that is both ergonomic and functional, with imaging optics and electronics housed in the 'underbelly' (see Figures 29, 30 and 31).

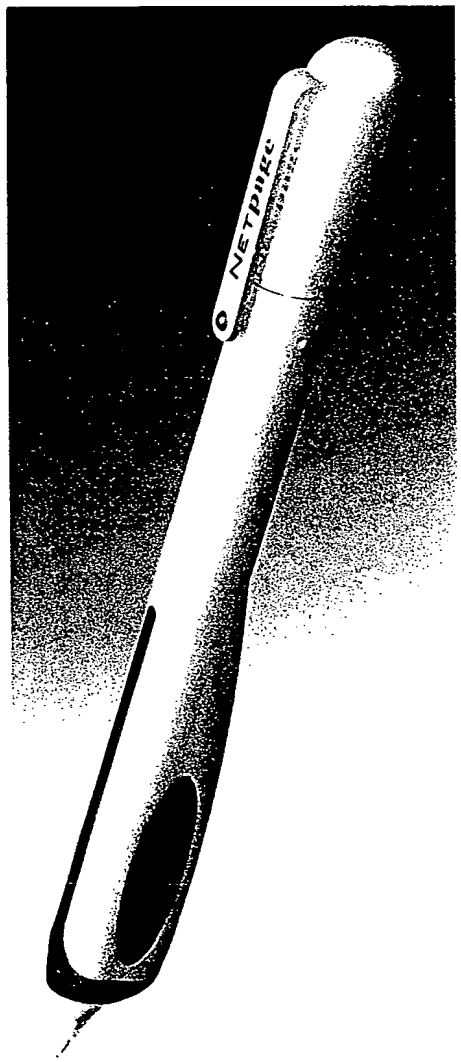


Figure 29. Netpage Pen, shown full size

The pen consists of a metal nib that is removable to allow ink cartridge replacement, followed by a tinted diffuser. The diffuser houses an antenna and two LEDs on a flex PCB; the first is located on top of the pen for good all-round visibility. It is a tri-color LED that responds in three modes when the pen is in use: periodic flashing green when it is online to the printer, momentary green when an operation succeeds, momentary red when an error occurs, yellow when it passes over an active area on the page. A separate lens is

mounted into the diffuser, which is optically de-coupled. The lens sits under a second LED that emits infrared light onto the page. The illuminated image is auto-focused onto a controller chip with an on-board image sensor via dedicated optics. The optics chassis accommodates a PCB with various components, including an induction coil for recharging and a MEMS chip. This chip also includes an optical sensor for detecting pressure movement in the metal cam turning when either the stylus or the ink cartridge is used for writing. The cam turning is connected to a terminal collar that has the contact strips for a rechargeable battery. This assembly is fitted into the pen top and connected via two flying leads to the PCB for power transmission. The top assembly is pushed into the body molding where it is free to rotate. By turning the top through 90-degree steps the pen has the stylus out, the pen out or both retracted. The pen has a standard length of 154mm and diameter of 11mm.

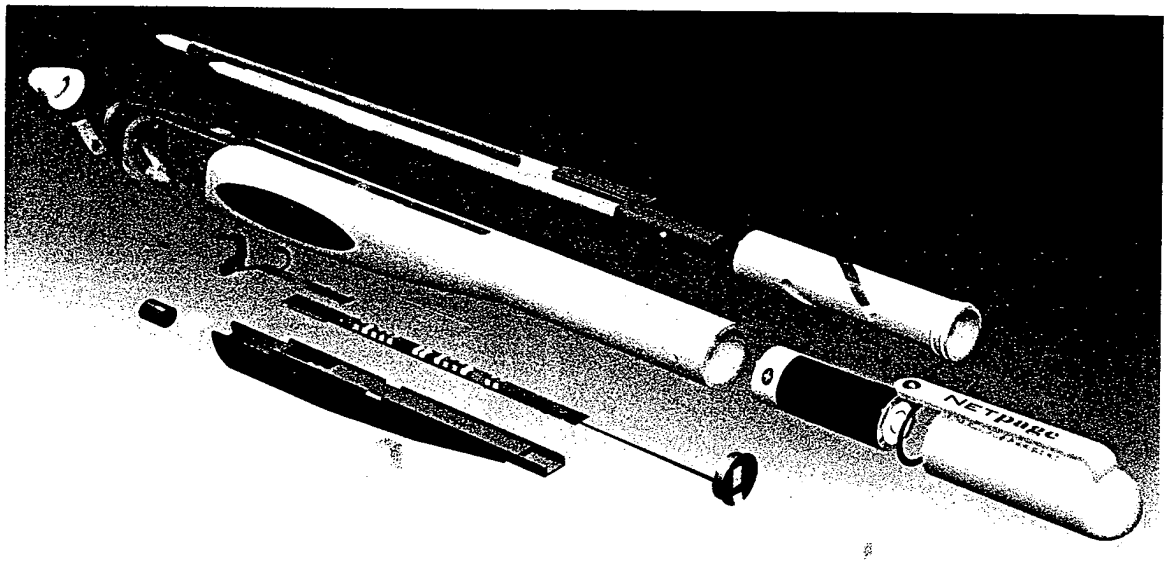


Figure 30. Netpage Pen, exploded view

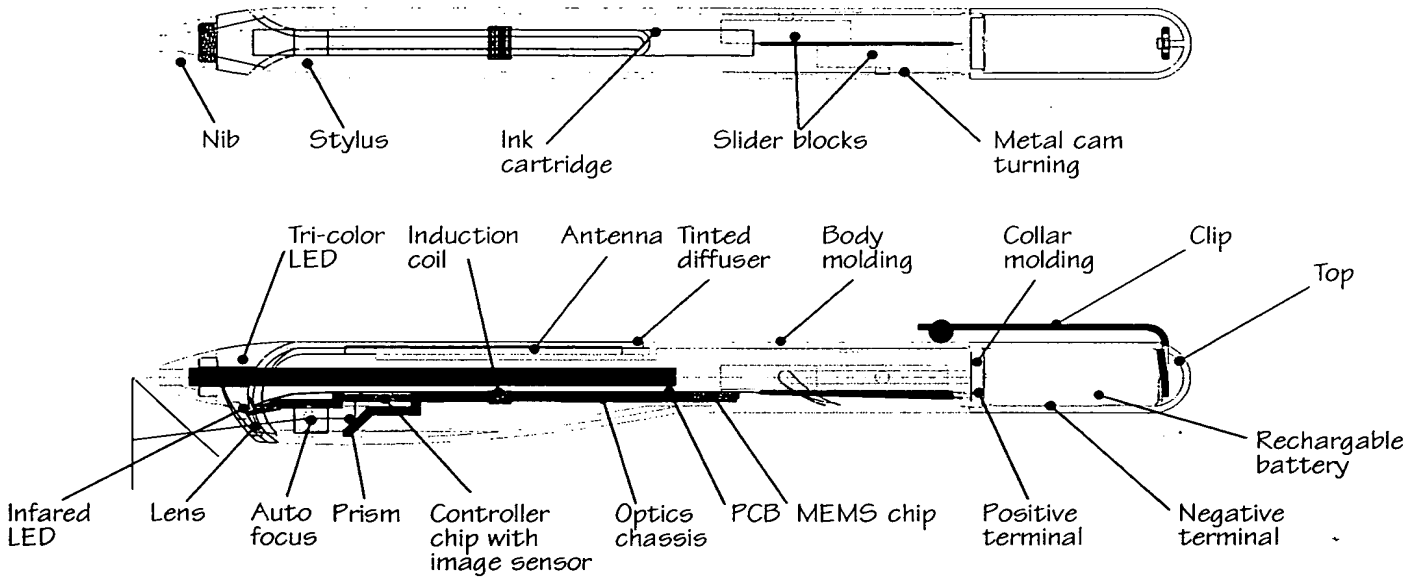


Figure 31. Netpage Pen

11.10 PEN CHARGING CUP

The pen charging cup is a simple device that simultaneously recharges any number of Netpage Pens that are placed in it (see Figure 32). Figure 33 shows the unit components which are an inner molding, base molding, an induction coil, a termination block, a LED, a diffuser molding and an exit wire to a 3V AC power transformer. The inner molding accommodates the induction coil, which is held captive when the unit is assembled. The coil passes through a termination block, where flying leads to a LED and series current-limiting resistors is also attached. The LED is positioned at the top of the unit under a diffuser, so it acts as a beacon to indicate that the charging cup is active. The charging cup is 70mm in diameter by 93mm high.

Figure 32. Pen charging cup and range of pens, and exploded view (inset)

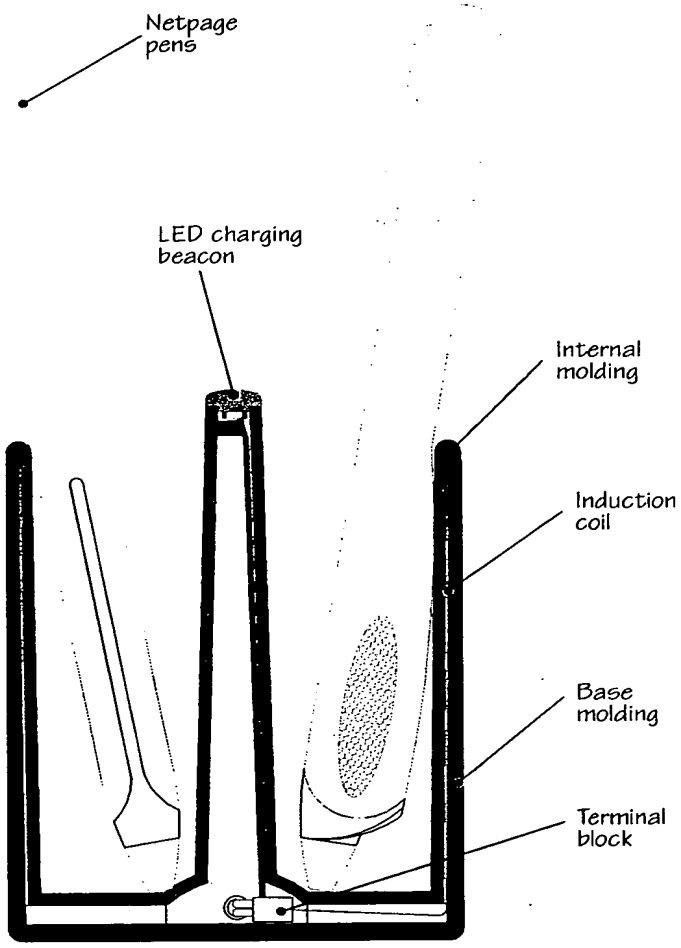


Figure 33. Netpage Pen charging cup

12 Memjet-Based Printing

A Memjet printhead produces 1600 dpi bi-level CMYK. On low-diffusion paper, each ejected drop forms an almost perfectly circular 22.5μm diameter dot. Dots are easily produced in isolation, allowing dispersed-dot dithering to be exploited to its fullest. Since the Memjet printhead is the width of the page and operates with a constant paper velocity, the four color planes are printed in perfect registration, allowing ideal dot-on-dot printing. Since there is consequently no spatial interaction between color planes, the same dither matrix is used for each color plane. Dot-on-dot printing minimizes 'muddying' of mid-tones caused by inter-color bleed.

A page layout may contain a mixture of images, graphics and text. Continuous-tone (contone) images and graphics are reproduced using a stochastic dispersed-dot dither. Unlike a clustered-dot (or amplitude-modulated) dither, a *dispersed-dot* (or frequency-modulated) dither reproduces high spatial frequencies (i.e. image detail) almost to the limits of the dot resolution, while simultaneously reproducing lower spatial frequencies to their full color depth, when spatially integrated by the eye. A *stochastic* dither matrix is carefully designed to be free of objectionable low-frequency patterns when tiled across the image. As such its size typically exceeds the minimum size required to support a particular number of intensity levels (e.g. 16×16×8 bits for 257 intensity levels).

Human contrast sensitivity peaks at a spatial frequency of about 3 cycles per degree of visual field and then falls off logarithmically, decreasing by a factor of 100 beyond about 40 cycles per degree and becoming immeasurable beyond 60 cycles per degree [31,40]. At a normal viewing distance of 12 inches (about 300mm), this translates roughly to 200-300 cycles per inch (cpi) on the printed page, or 400-600 samples per inch according to Nyquist's theorem.

In practice, contone resolution above about 300 ppi is of limited utility outside special applications such as medical imaging. Offset printing of magazines, for example, uses contone resolutions in the range 150 to 300 ppi. Higher resolutions contribute slightly to color error through the dither.

Black text and graphics are reproduced directly using bi-level black dots, and are therefore not antialiased (i.e. low-pass filtered) before being printed. Text is therefore *supersampled* beyond the perceptual limits discussed above, to produce smoother edges when spatially integrated by the eye. Text resolution up to about 1200 dpi continues to contribute to perceived text sharpness (assuming low-diffusion paper, of course).

The Netpage Printer uses a contone resolution of 267 ppi (i.e. 1600 dpi / 6), and a black text and graphics resolution of 800 dpi.

13 Document Data Flow

Because of the page-width nature of the Memjet printhead, each page must be printed at a constant speed to avoid creating visible artifacts. This means that the printing speed can't be varied to match the input data rate. Document rasterization and document printing are therefore decoupled to ensure the printhead has a constant supply of data. A page is never printed until it is fully rasterized. This is achieved by storing a compressed version of each rasterized page image in memory.

This decoupling also allows the RIP to run ahead of the printer when rasterizing simple pages, buying time to rasterize more complex pages.

Because contone color images are reproduced by stochastic dithering, but black text and line graphics are reproduced directly using dots, the compressed page image format contains a separate foreground bi-level black layer and background contone color layer. The black layer is composited over the contone layer after the contone layer is dithered.

Figure 34 shows the flow of a Netpage Printer document from network to printed page.

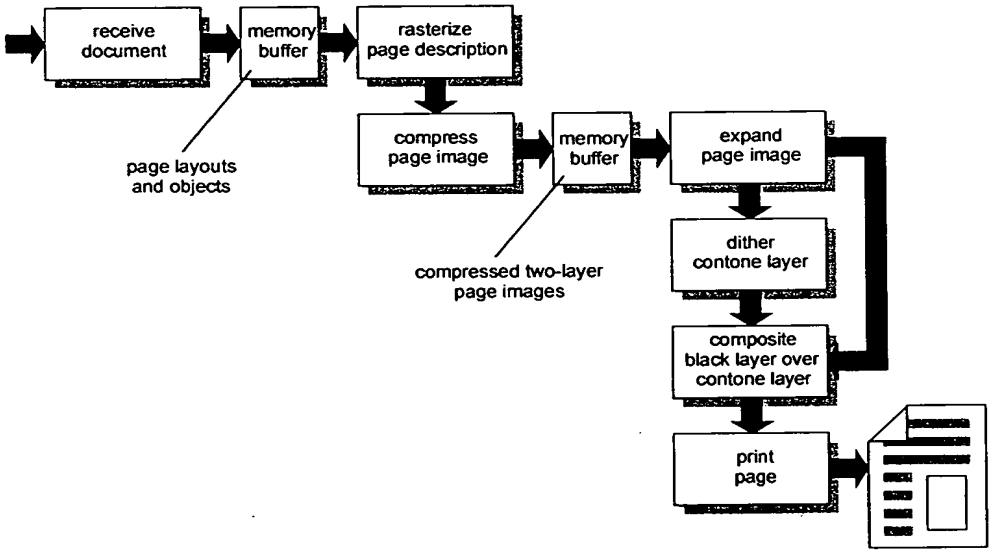


Figure 34. Netpage Printer document data flow

At 267 ppi, a Letter page of contone RGB or CMY data has a size of 19MB. Using lossy contone compression algorithms such as JPEG [45], contone images compress with a ratio up to 10:1 without noticeable loss of quality, giving a compressed page size of 1.9MB.

At 800 dpi, a Letter page of bi-level data has a size of 7MB. Coherent data such as text compresses very well. Using lossless bi-level compression algorithms such as Group 4 Facsimile [7], ten-point text compresses with a ratio of about 10:1 (as discussed in Section 18.2.1.2), giving a compressed page size of 0.8MB.

Once dithered, a page of CMY contone image data consists of 86MB of bi-level data. Using lossless bi-level compression algorithms on this data is pointless precisely because the optimal dither is stochastic - i.e. since it introduces hard-to-compress disorder.

The two-layer compressed page image format therefore exploits the relative strengths of lossy JPEG contone image compression and lossless bi-level text compression. The format is compact enough to be storage-efficient, and simple enough to allow straightforward real-time expansion during printing.

Since text and images normally don't overlap, the normal worst-case page image size is 1.9MB (i.e. image only), while the normal best-case page image size is 0.8MB (i.e. text only). The absolute worst-case page image size is 2.7MB (i.e. text over image). Assuming a quarter of an average page contains images, the average page image size is 1.1MB.

14 Printer Controller Architecture

The Netpage Printer controller consists of a controlling processor, a factory-selected network interface, a radio transceiver, dual raster image processor (RIP) DSPs, duplexed print engines, flash memory, and 64MB of DRAM, as illustrated in Figure 35.

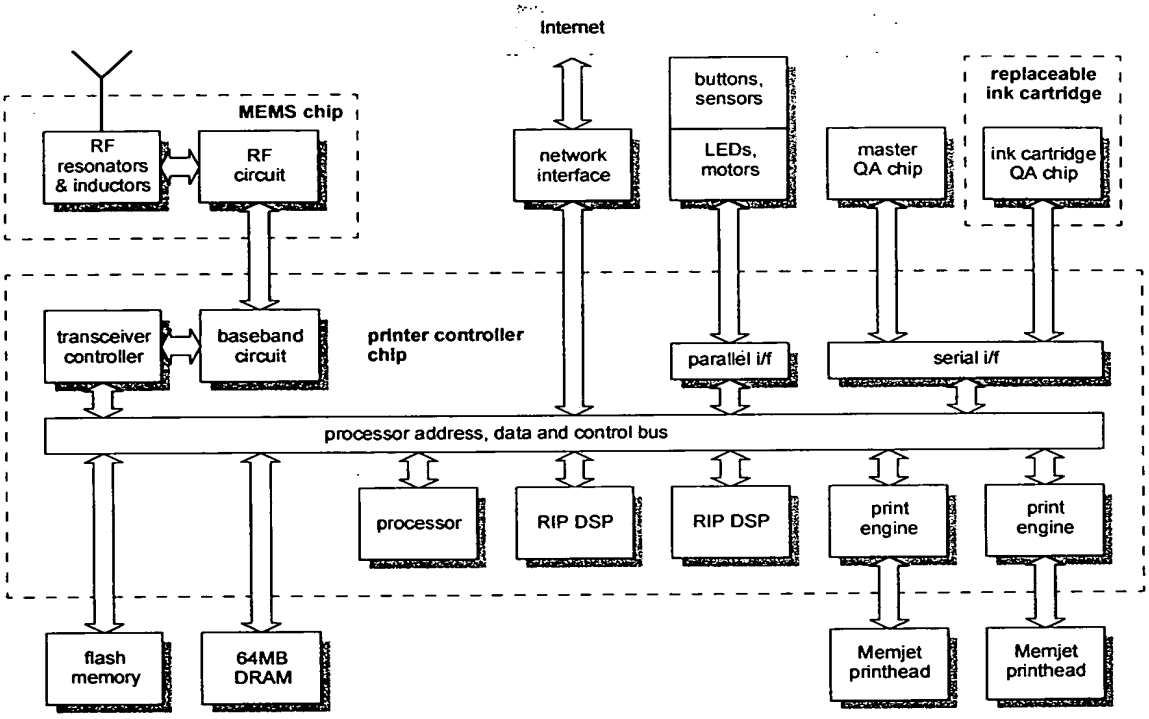


Figure 35. Basic printer controller architecture

The controlling processor handles communication with the Internet and with local wireless pens, controls the user interface (buttons and LEDs), controls the paper transport, handles ink cartridge authentication and ink monitoring, and feeds and synchronizes the RIP DSPs and print engines. It consists of a medium-performance general-purpose microprocessor.

The RIP DSPs rasterize and compress page descriptions to the Netpage Printer's compressed page format. Each print engine expands, dithers and prints page images to its associated Memjet printhead in real time (i.e. at 30 or 45 pages per minute). The duplexed print engines print both sides of the page simultaneously.

The printer controller's flash memory holds the software for both the processor and the DSPs, as well as configuration data. This is copied to main memory at boot time.

The processor, DSPs, print engines and digital transceiver components are integrated in a single ASIC. The MEMS and analog RF components are integrated in a separate MEMS chip, which is also used in the wireless pen. Additional pen-specific components in the

MEMS chip are not used in the printer controller. The Internet network interface module is separate, since Netpage Printers allow the network connection to be factory-selected. Flash memory and the 2×256Mbit (64MB) DRAM is also off-chip.

Various Internet network interface modules can be supported. Possibilities include a POTS modem, a Hybrid Fiber-Coax (HFC) cable modem, an ISDN modem, a DSL modem, a satellite transceiver, a current or next-generation cellular telephone transceiver, a wireless local loop (WLL) transceiver, etc. A Internet connection may already be available on the local network, in which case only a local network connection may be required.

The printer controller optionally includes a local network connection, to allow the printer to be used from a directly-connected workstation or over a local-area network. Possibilities include 10Base-T and 100Base-T Ethernet, USB and USB 2.0, IEEE 1394 (Firewire), and various emerging home networking standards.

The radio transceiver communicates in the unlicensed 900MHz band normally used by cordless telephones, and uses frequency hopping and collision detection to provide interference-free communication.

14.1 Detailed Document Data Flow

The main processor receives and verifies the document's page layouts and page objects by Internet pointcast and multicast as described in Section 5.1. It then runs the appropriate RIP software on the DSPs.

The DSPs rasterize each page description and compress the rasterized page image. The main processor stores each compressed page image in memory. The simplest way to load-balance multiple DSPs is to let each DSP rasterize a separate page. The DSPs can always be kept busy since an arbitrary number of rasterized pages can, in general, be stored in memory. This strategy can lead to poor DSP utilization, however, when rasterizing short documents.

Watermark regions in the page description are rasterized to a contone-resolution bi-level bitmap which is losslessly compressed to negligible size and which forms part of the compressed page image.

The infrared (IR) layer of the printed page contains encoded position tags at a density of about 25 per inch. Each tag encodes the page id, tag position, and control bits. Active areas and pressure-sensitive areas in the page description are rasterized to tag-resolution bi-level bitmaps which do not require compression and which form part of the compressed page image.

The main processor passes back-to-back page images to the duplexed print engines. Each print engine stores the compressed page image into its local memory, and starts the page expansion and printing pipeline. Page expansion and printing is pipelined because it is impractical to store a 114MB bi-level CMY+IR page image in memory.

The first stage of the pipeline expands the JPEG-compressed contone CMY layer, expands the Group 4 Fax-compressed bi-level watermark map, and expands the Group 4 Fax-compressed bi-level black layer, all in parallel. The second stage dithers the contone CMY layer using the dither matrix selected by the watermark map, and composites the bi-level black layer over the resulting bi-level CMY layer. Since there is no black ink used in the Netpage Printer, the black layer is composited with each of C, M and Y. In parallel with

this, the tag encoder generates and encodes the bi-level IR tag data. The last stage prints the bi-level CMY+IR data through the Memjet printhead via the printhead interface.

14.2 PRINT ENGINE ARCHITECTURE

The print engine's page expansion and printing pipeline consists of a standard JPEG decoder, a standard Group 4 Fax decoder, a custom halftoner/compositor unit, a custom position tag encoder, and a custom interface to the Memjet printhead. These are described in detail in Section 19.

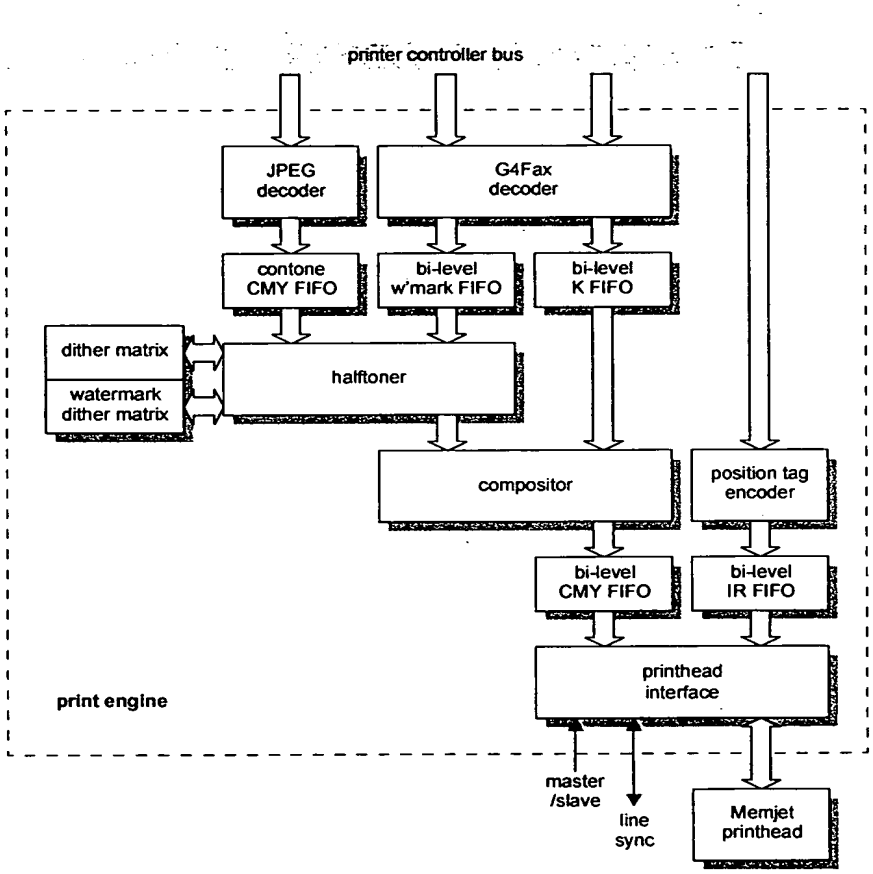


Figure 36. Print engine architecture

When several print engines are used in unison, such as in a duplexed configuration, they are synchronized via a shared line sync signal. Only one print engine, selected via the external master/slave pin, generates the line sync signal onto the shared line.

In the 8½" versions of the Netpage Printer, the two print engines each prints 30 Letter pages per minute along the long dimension of the page (11"), giving a line rate of 8.8 kHz at 1600 dpi. In the 11" Pro versions of the Netpage Printer, the two print engines each prints 45 Letter pages per minute along the short dimension of the page (8½"), giving a line rate of 10.2 kHz. These line rates are well within the operating frequency of the Memjet printhead, which in the current design exceeds 30 kHz.

15 Pen Controller Architecture

The Netpage Pen controller consists of a controlling processor, a radio transceiver, a tilt sensor, a nib pressure sensor, a IR image sensor, flash memory, and 512KB of DRAM, as illustrated in Figure 37.

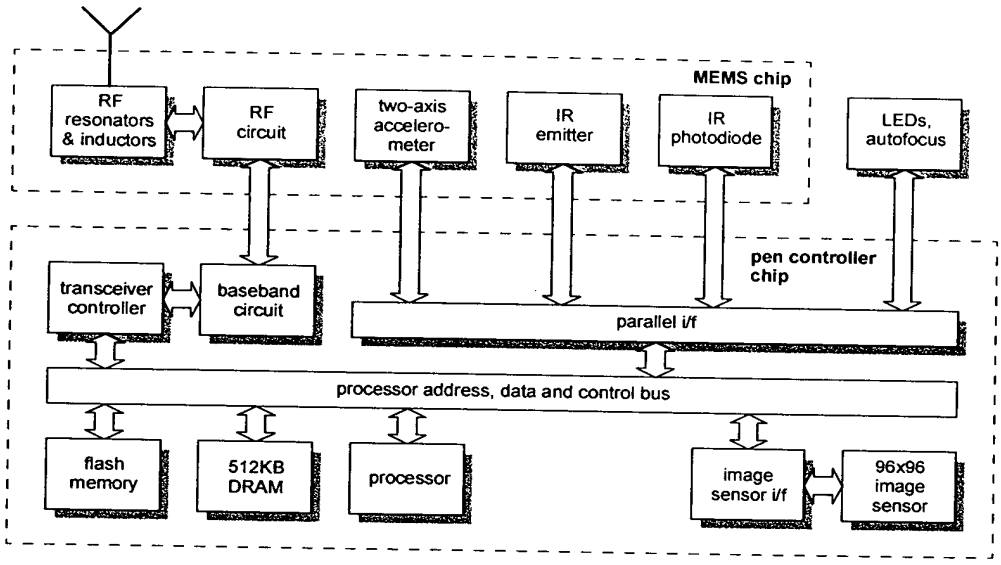


Figure 37. Pen controller architecture

The controlling processor captures and decodes IR position tags from the page via the image sensor, monitors the tilt and pressure sensors, controls the autofocus voice-coil, controls the user interface (tricolor LED), and handles wireless communication with the local Netpage Printer. It consists of a medium-performance (~25MHz RISC) general-purpose microprocessor.

Two-axis tilt sensing is provided by a two-axis accelerometer. Nib pressure sensing is provided by an IR emitter and photodiode pair in conjunction with a reflector coupled with the sprung nib.

The processor, digital transceiver components, 96×96 image sensor, flash memory and 512KB DRAM are integrated in a single ASIC. The MEMS and analog RF components, accelerometers, and the IR emitter/photodiode are integrated in a single MEMS chip, also used in the Netpage Printer.

The radio transceiver communicates in the unlicensed 900MHz band normally used by cordless telephones, and uses frequency hopping and collision detection to provide interference-free communication.

NETPAGE ENCODING AND DECODING

16 Netpage Encoding

As described in Section 5.2, hidden information on a printed Netpage encodes the unique id of the page instance, together with position information which tells the pen which part of the page the user is interacting with. The Netpage Network can correlate any reference to any part of any page ever printed with the contents of the page, solely on the basis of the page id and the position.

The entire page is tiled with position tags. Each position tag must be independent, to allow for single-point virtual button and hyperlink presses, and so contains both the local x-y position and the global page id. In addition particular flags are associated with each tag, such as whether the pen should turn on its light while over the tag.

The tag data (page id, a local x-y position, and control flags) is error-correctably encoded as a square binary matrix which is printed as an array of bi-level macrodots. Each macrodot itself consists of a square of one or more printed dots to ease image processing.

The position encoding supports a particular page size. Where the physical page size exceeds this virtual page size, the physical page is simply tiled with multiple virtual pages, each with a different page id. Figure 38 illustrates the concept of a page tiled with position tags. The contents of each tag are not shown on this illustration.

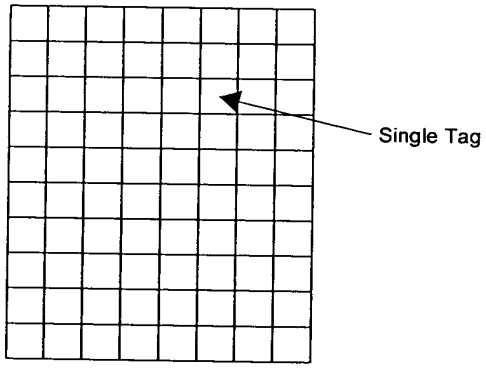


Figure 38. Part of a page tiled with position tags

The tags are printed with an infrared-absorptive ink that can be read by the pen device. Since black ink is IR absorptive, limited functionality can be provided on offset-printed pages using black ink on otherwise blank areas of the page - for example to encode buttons. Alternatively an invisible infrared ink can be used to print the position tags over the top of a regular page. However, if invisible IR ink is used, care must be taken to ensure that any other printed information on the page is printed in infrared-*transparent* CMY ink, for black ink will obscure the infrared tags. The monochromatic scheme was chosen to maximize dynamic range in blurry reading environments.

Simple virtual button presses don't specifically require invisible ink. Buttons could be visibly coded since the real estate is available. Hyperlinks require invisible ink, but impose no other requirements on the tag encoding.

Table 4. Encoding parameters

parameter	definition	value
p	virtual page width (inches)	12[a]
d	dot pitch (dots per inch)	1600
k	macrodot width (dots)	4
b	registration border width (macrodots)	2
r	data redundancy factor	1.5
m	coordinate precision (bits)	-
g	page id precision (bits)	-
s	tag width (macrodots)	-
n	tags per page[b]	-

a. US Letter page length
b. per dimension

Given the encoding parameters defined in Table 4, the tag width is given by:

$$s = b + \sqrt{r(2m + g)}$$

Given a particular tag width, the number of tags per page in each dimension is given by:

$$n = \frac{pd}{ks}$$

This in turn yields the required coordinate precision:

$$m = \log_2 n$$

If 10 billion people all generate 100,000 pages per annum for 1000 years, they will generate 10^{18} pages, or approximately 2^{60} pages. A page id precision of 64 bits should therefore be sufficient, notwithstanding issues of efficient contiguous page id allocation.

Assuming a page id precision of 64 bits, and other parameter values as given in Table 4, the equations converge on a tag width of 17 macrodots, 283 tags per page dimension, and a coordinate precision of 9 bits. This precision supports a maximum page dimension of 21.7" or 552mm.

Given a particular tag width, the position resolution, *in millimeters*, is given by:

$$\frac{25.4ks}{d}$$

and, *in points*, is given by:

$$\frac{72ks}{d}$$

This yields a position resolution of 3 points (1.08mm at 1600 dpi).

Magazine-quality printed text normally has a size of 10 points. Forms filled in by hand normally allow for handwritten text with a size of about 20 points. A position resolution of 3 points therefore translates to a character-oriented resolution of 3.3 for printed text and 6.6 for handwriting recognition. This position resolution is increased to the 200 dpi required for handwriting recognition by taking into account the position of the tag within the captured image area.

With a position tag width of 17 macrodots, and 4×4 dots (at 1600 dpi) per macrodot, the raw position tag gives about 24 positions per inch. Consequently any positioning scheme must achieve 10 times better resolution based on tag position within the sensed image.

16.1 Position Tag Structure

Each position tag encodes the following information:

Table 5. Data Encoded in a Position Tag

Name	#Bits	Description
PageId	64	Defines the Netpage page instance id
x	9	Defines the x coordinate within the given PageId
y	9	Defines the y coordinate within the given PageId
ActiveArea	1	Defines whether the pen should turn its light on while over the tag
PressureSensitiveArea	1	Determines whether the pen should return continuous pressure or not while over the tag. 0 = don't return pressure readings. 1 = return pressure readings
Reserved	6	For future use. Store as 0 for this version.
TOTAL	90	

Each position tag is a 17×17 array of macrodots, with each macrodot being a 4×4 square of 1600 dpi dots, as shown Figure 39.

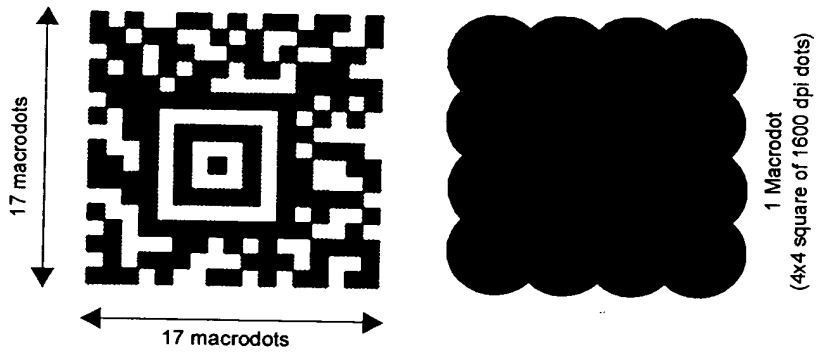


Figure 39. Macrodots in a Position Tag

Since the dots are monochrome IR-absorptive dots, printed on an IR-reflective background, a "black macrodot" is physically different from a "white macrodot". Figure 40 illustrates a magnified view of macrodots.

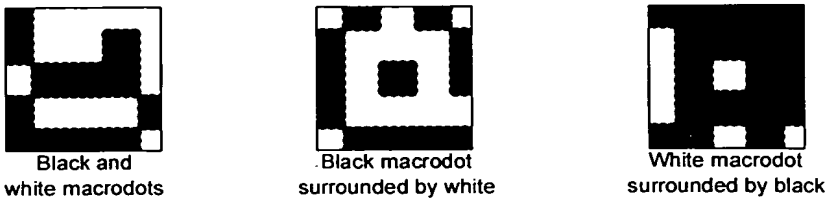


Figure 40. Magnified Macrodots

The position tag structure is based on concepts used in the public-domain Aztec 2D Barcode [90], invented by Andy Longacre of Welch Allyn Inc. in 1995. While the position tag uses the same bull's-eye structure mechanism from the Small Aztec Symbol for symbol location and orientation, we define our own interpretation for the mode bits and do not allow variable length structures. In addition, the position tag is a 17×17 array, which is not possible in a pure Aztec code.

Figure 41 shows the high-level structure of a position tag in relation to its 17×17 macrodot array. The target/orientation area consists of a 5-layer bull's-eye with 3-bit orientation markers on the outer corners. The format of this area is identical to that used in the Small Aztec Barcode although it is placed off-center. The Mode Data Area is the area between the orientation markers in what would otherwise be the 6th layer of the bull's-eye target. The remaining area is used to hold the actual data for the position tag.

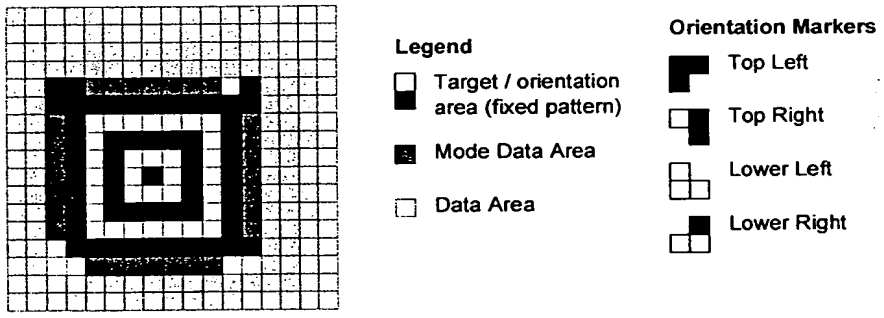


Figure 41. High Level Structure of a Position Tag

16.1.1 Target and Orientation Area

The target and orientation area consists of the fixed pattern shown in black and white in Figure 41. The format of this area is identical to that used in the Small Aztec Barcode.

The black positions in are black macrodots, and the white positions are white macrodots. The 5-layer bull's-eye pattern is easily found in a 2D image by scans for topological connectivity, and is then useful for pinpointing the exact center regardless of orientation, and for determining the main axes and local x-y dimensions. On the four corners of the target are 3-bit clusters of orientation bits, which allow the tag's orientation to be quickly determined.

16.1.2 Mode Data Area

The bits in the layer immediately adjoining the target (other than the orientation bits) comprise a 28-bit string, starting upper left and circling clockwise. This area is known as the Mode Data Area. The relative position of the 28 Mode Data bits can be seen in Figure 42.

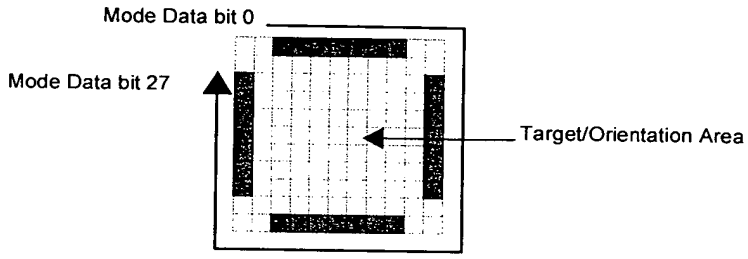


Figure 42. Mode Data Area

The 28-bit string encodes meta-information used to decode the remainder of the position tag. Of particular note is the first bit of the meta information. If this bit is 0, then the decoding of the data area is as described here. Setting this bit to 1 allows for redefinition of the position tags at some later date while remaining compatible with existing pages. The meta information and its encoding to 28 bits is described in Section 16.2.1.

16.1.3 Data Area

When the first bit of the meta information decoded from the Mode Data Area is 0, then the data area is defined to be the 168 bits as shown in Figure 43.

The encoded message data is placed onto the tag in 2-cell high layers, starting at the upper left and spiraling clockwise out to the edge of the symbol. The data is therefore interpreted as a sequence of related "dominos", each 1 wide and 2 tall, with their more significant bit always further from the target area (a black macrodot represents a "1" bit, and a white macrodot represents a "0" bit). Figure 43 illustrates the sequence and orientation of the dominos when turning the corners in the data area.

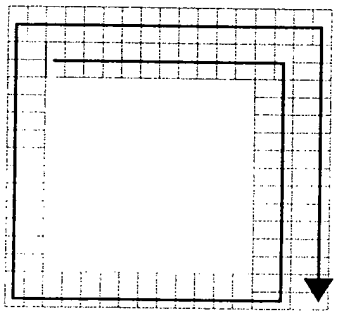


Figure 43. Domino Mapping and Sequencing of Data Area

The systematic domino-based layout of the data simplifies both encoding and decoding at the graphical level. The encoding mechanism for the data is described in Section 16.2.2, although it is worth noting here that decoding errors and tag damage are expected to occur more towards the edges of the tag.

Note that the 17×17 structure of the position tag means that with reference to the strict Aztec Code, the first layer of data is complete, but the second layer of data is only half full. If the second layer was completely full the position tag would be a 19×19 square cell with the target area in the center.

16.2 POSITION TAG ENCODING

16.2.1 The Mode Message

The 28-bit string in the Mode Area is used to hold 8 mode bits. These 8 mode bits describe how the remainder of the position tag bits are encoded.

The 8 mode bits are encoded into 28 bits via Reed-Solomon encoding. The Mode Area therefore holds the 8 mode bits and 20 additional check bits. The 8 mode bits are parsed into two 4-bit words, and then 5 additional check words are computed by systematic Reed-Solomon encoding over the Galois field GF(16) based on a prime modulus polynomial of: $x^4 + x + 1$ (=19 decimal). The generator polynomial of $(x\text{-}2^1)..(x\text{-}2^4)$ is:

$$x^4 + 11x^3 + 4x^2 + 6x + 2$$

Reed-Solomon encoding is chosen for its ability to deal with burst errors and effectively detect and correct errors using a minimum of redundancy. Reed-Solomon coding is discussed in detail in [91], [71], and [54]. The reader is advised to refer to these sources for background information.

Table 6 shows the interpretation of the mode bits. Note that the first bit of the mode data is a version bit, and determines the interpretation of the remainder of the mode bits and the tag data. The Version bit should therefore be set to 0. Later interpretations of the tag can be defined at a later date via the version bit.

Table 6. Interpretation of Mode Bits

#Bits	Name	Description
0	Version	0 = the interpretation of the remaining 7 bits and the data area is as described in this document 1 = reserved for future use
1-2	NumMsgWords	Defines how many 6-bit codewords of the data area are message codewords, and how many are check-word codewords. This number is determined by the encoding scheme used for data (see Section 16.2.2). Values are: 00 = 15 data message codewords 01 = 16 data message codewords 10 = 17 data message codewords 11 = 18 data message codewords
3-7	Reserved	0

16.2.2 The Data Message

The Data Area contains 168 raw bits, which is an encoded form of the 90 bits of position tag data (see Table 5). The 168 bits are represented by 104 bits in the first layer surrounding the target area and 64 bits in the half layer surrounding the first layer.

The 90 bits of position tag data are encoded into 168 bits in two steps. The first step breaks the 90 bits into codewords of 6 bits such that there are no codewords with all 0s or all 1s.

The second step Reed-Solomon encodes the data over the Galois field GF(64), which generates additional 6-bit check words.

16.2.2.1 Step 1: Message Encoding

In the first step of message encoding, the 90 data bits are arranged into a sequence of 6-bit message codewords in a generally direct fashion, starting at the most significant bit of the first codeword, with two key exceptions:

- whenever the first 5 bits placed in a codeword are all "0"s, then a dummy "1" is inserted into that codeword's LSB and the following message bit starts off the next codeword.
- whenever the first 5 bits placed in a codeword are all "1"s, then a dummy "0" is inserted into that codeword's LSB and the following message bit starts off the next codeword.

In the end, the character and byte boundaries in the original 90-bit message have no necessary relationship with the codeword boundaries. Up to 5 bits may remain unfilled in the final message codeword, and they are padded out with "1"s (and possibly a final dummy "0" if necessary) to eliminate any ambiguity.

The code-forming rules are designed to never create a *message* codeword of all 0s or all 1s, but the error encoding adds on additional codewords of any value. Thus during decoding, an occurrence of those illegal values within the message region (but *not* within the check region) can be regarded as a correctable *erasure* (this is more useful than not knowing which codeword is in error).

In the case of our 90-bit original messages, there are two cases to consider:

- the *best* case is that 15 codewords are generated for the data message
- the *worst* case is that 18 codewords are generated for the data message

The codeword based message therefore consumes D 6-bit codewords, where D is in the range 15-18.

16.2.2.2 Step 2: Generate Check Words

The second step of message encoding involves generating the check words for the data message. Since the data message consumes between 15 and 18 6-bit codewords of the 28 available codewords, the remaining 13 to 10 codewords can be used for check words.

The K additional check words are computed by systematic Reed-Solomon encoding over the Galois field GF(64) based on a prime modulus polynomial of: $x^6 + x + 1$ (= 67 decimal). The generator polynomial is simply expanded out to $(x\text{-}2^1)..(x\text{-}2^K)$ as needed at the time of printing. The check words are not adjusted in the same way as the message bits are adjusted.

16.2.2.3 Step 3: Place into Data Area

The final message to be stored in the data area of a position tag consists of 28 6-bit codewords made up of D message codewords followed by K check codewords. The exact number of message codewords, D, is stored into the 2-bit NumMsgWords component of the Mode information for the position tag.

The resulting sequence of codewords is parsed anew into a sequence of 2-bit dominos (see Figure 44) and is then taken in reverse order and graphically arranged spiralling clockwise and outwards through the data area (see Figure 43). As decoding errors and tag damage are expected to occur more towards the edges of the tag, by reversing the codeword sequence, the message codewords occupy those edge regions.

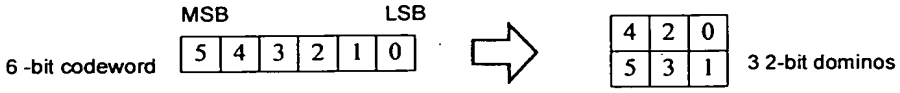


Figure 44. Relationship of 6-bit codeword to generated dominos

One macrodot is placed into the data area per bit. The placement of an IR-absorptive ink macrodot represents a "1". If there is no IR-absorptive ink placed on the page at the particular location, then that macrodot represents a "0".

17 In-Pen Processing

The pen must send the printer 100 samples per second. The mechanism for reading positional information from the position tags must work at this 100 Hz rate. The following tasks must be performed each sample:

- capture time, tilt and pressure
- capture the image
- locate the position tag
- decode the position tag
- build stroke information
- activate the light as required
- encrypt the stroke data
- transmit the encrypted data

When a pen-down is registered, we capture the first three frames at the two extreme focus values and the middle focus value. After that we perform regular auto-focus based on a sharpness metric. In this way we are more likely to capture a "single-click" event.

While the pen is in range of the printer, partially completed strokes are transmitted as they are being formed. When the pen moves out of range, stroke information is buffered within the pen (approximately 12 minutes of pen motion while physically on the page), and transmitted later.

Each pen contains a characteristics block which can be read by the printer. It contains at least the information shown in Table 7.

Table 7. Pen characteristics block

Name	Description
PenId	A unique pen identifier - unique across all pens
ModelId	The model number of this pen[a]
ManufactureId	The manufacturer of this pen
TiltInfo	The relationship between tilt and pen position.

a. This can be combined with ManufactureId to allow the printer to download a set of pen characteristics. However it may be more convenient to store the characteristics (such as TiltInfo) in the pen.

17.1 Position Tag Decoding

17.1.1 Capture Time, Tilt and Pressure

The pen contains sensors which allow pen tilt and nib pressure to be determined.

17.1.1.1 Time

All measurements are made in the context of a particular timestamp. The pen contains a timer to allow equally time-spaced measurements to be made. Although we capture samples at a rate of 100 per second, we report time at millisecond resolution to allow for future improvements to the pen. Time resolution is provided at 32 bits.

17.1.1.2 Tilt

The pen contains two orthogonal passive accelerometers in the x-y plane perpendicular to the pen axis. These respond to gravity and allow two-axis tilt to be computed to an accuracy of at least 5 bits.

Tilt is required to determine the nib position. The printer uses the two tilt values and the pen's non-corrected pen position to determine the actual pen nib position. The Pen Characteristics Block contains the relationship required for the calculation. Offloading the correction calculation to the printer reduces pen complexity and price.

17.1.1.3 Pressure

The pen contains a pressure sensor which measures to an accuracy of at least 5 bits. From the pen's point of view, the pressure value is only used to determine whether the pen is on the paper or not. The printer is responsible for all other interpretations of pressure, such as determining whether an item on a page is being selected. The pen merely passes the pressure value on to the printer.

Each position tag contains a data bit called PressureSensitiveArea. If a given pen stroke includes position tags that have their PressureSensitiveArea bit set, then the pressure value is passed as part of the stroke information. If the PressureSensitiveArea is clear, then pressure is implied from a stroke in terms of pen-down and pen-up.

17.1.2 Capture the Image

The image needs to be captured for the pen position to be determined. This only needs to happen if the pen is on the page.

To capture an image, data must be transferred from the image sensor to a fixed image buffer. This process involves a single read and write per pixel. The minimum and maximum values encountered during the transfer are kept in variables MinPixelEncountered and MaxPixelEncountered, respectively, for later use in the data recovery process (see Section 17.1.4).

There are a number of general considerations that are part of the assumptions for reading in and decoding a position tag from an image sensor.

With regards to the image sensor itself, there are two calculations to consider: the size of the sensed area, and the resolution of the image sensor. As described in Section 17.1.2.1 and Section 17.1.2.2, the sensed area has a required size of 193×193 dots (or 3.06mm × 3.06mm), and the image sensor has a required resolution of 96×96 pixels.

Furthermore, as discussed in Section 17.1.2.3, the sensed image may be blurry, and the sense of ambiguous macrodots may have to be inferred.

A 96×96 image contains 9216 8-bit pixels. We round this up to 10,000 pixels for calculation purposes. The transfer process involves on average 6.5 cycles per pixel for a total of 65,000 cycles per position tag. On a 25 MHz processor this represents 26% of the available bandwidth.

17.1.2.1 Sensed Area Size

The basic position tag consists of 17×17 macrodots. Since each macrodot is a 4×4 array of printed dots, each position tag in fact consists of 68×68 printed dots.

To ensure that one entire tag is always within the window of the image sensor, the image sensor window must be the width of two tags. This requirement increases the sensed dot area to 136×136.

Finally, since the pen must detect positional tags at any rotation, we must consider the worst case rotation of 45 degrees. This requirement increases the sensed dot area to 193×193.

At a printed dot size of 1600 dpi, the image sensor must sense a printed area containing 193×193 dots, which equates to 3.06mm (0.12 inches) in each dimension.

17.1.2.2 Image Sensor Resolution

The basic positional tag is 17×17 macrodots. This equates to a basic sensor resolution of 17×17 pixels.

To ensure that one entire tag is always within the window of the image sensor, the image sensor window must be the width of two tags. This requirement increases the sensor resolution to 34×34.

Since the pen must detect positional tags at any rotation, we must consider the worst case rotation of 45 degrees. This requirement increases the sensor resolution to 48×48.

Finally, to satisfy Nyquist's Theorem with respect to macrodots, we must oversample at least at twice the macrodot resolution. This leads to a sensor resolution of 96×96.

Each sampled pixel is 1 byte (8 bits). We must also assume the least significant 2 bits of each pixel can contain noise. Decoding algorithms must therefore be noise-tolerant.

17.1.2.3 Blurry Image

The pen optics provide reasonable depth-of-field, and the pen includes an auto-focus mechanism to handle changes in pen-paper separation due to varying tilt etc., but several factors may still contribute to blurriness in the captured image, including tardy auto-focus response, and varying pen-paper separation due to extreme tilt or warped paper.

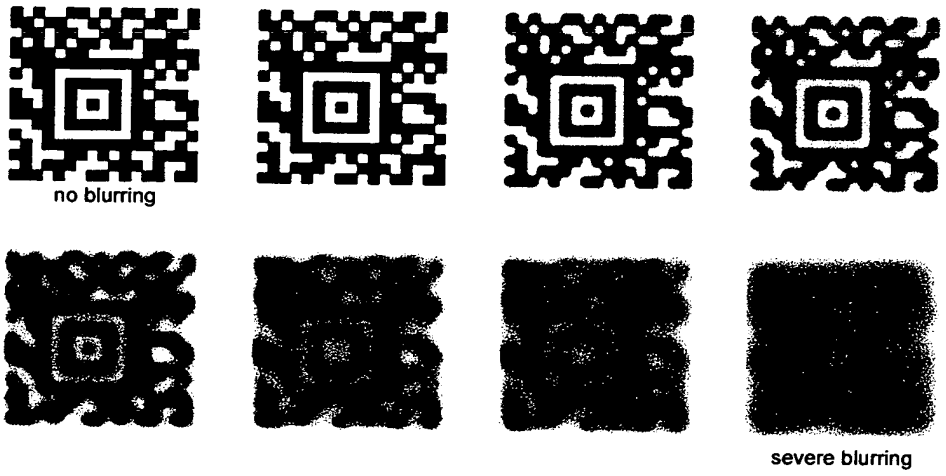


Figure 45. Effect of different degrees of blurring

Blurring can prevent definitive macrodot recognition. Figure 45 shows the effect of blurring on a position tag composed of macrodots, ranging from no blurring to severe blurring. Note how isolated macrodots fade as the blurring becomes more severe.

As the blurring increases, the more a given macrodot is influenced by the surrounding macrodots. Consequently the dynamic range for a particular macrodot decreases. Consider a "white" macrodot and a "black" macrodot, each surrounded by all possible sets of macrodots. The 9 macrodots are blurred, and the center macrodot sampled. Figure 46 shows the distribution of resultant center macrodot pixel values for black and white macrodots.

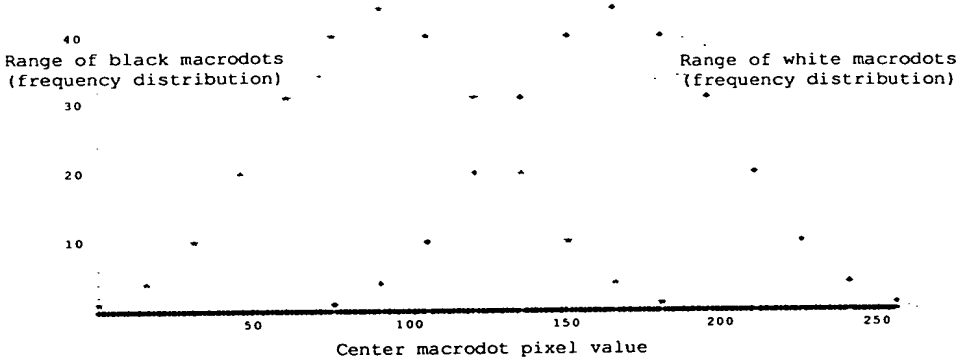


Figure 46. Black-white macrodot ambiguity due to blurring

The curve in Figure 46 is computed for an illustrative degree of blurring. The curve from pixel values 0-180 shows the range of black macrodots. The curve from pixel values 75-250 shows the range of white macrodots. The greater the degree of blurring, the more the two curves shift towards the center of the range and therefore the greater the intersection area, which means the more difficult it is to determine whether a given macrodot is black or white. A pixel value at the center point of intersection is ambiguous - the macrodot is equally likely to be black or white.

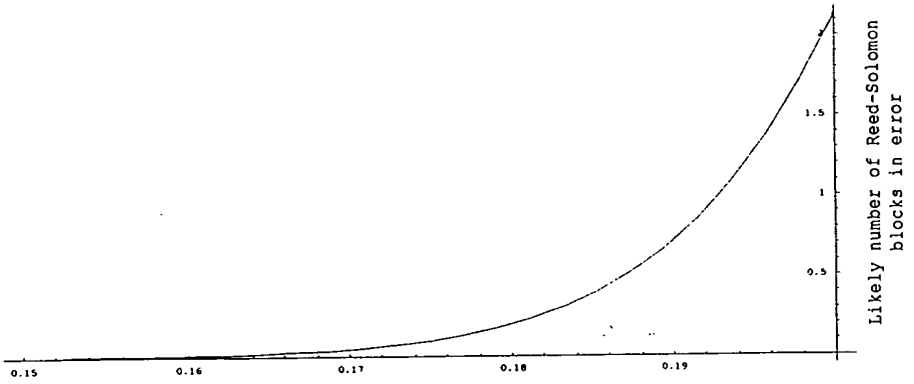


Figure 47. Relationship between data non-recovery and codeword error rate

As the blurring increases, the likelihood of a read bit error increases. Fortunately, the Reed-Solomon decoding algorithm can cope with bit errors gracefully up to t symbol errors. Figure 47 shows the predicted number of Reed-Solomon codewords that cannot be recovered given a particular symbol error rate. Notice how the Reed-Solomon decoding

scheme performs well and then substantially degrades. If there is no Reed-Solomon block duplication, then only 1 codeword needs to be in error for the data to be unrecoverable. Of course, with the inclusion of erasure detection (via invalid codewords), the chance of correctly decoding the data increases.

Figure 47 only links codeword errors to the total number of Reed-Solomon codewords in error. There is a trade-off between the amount of blurring that can be corrected, and the number of genuine errors due to dirt and paper damage that can be corrected. Since all error detection and correction is performed by the Reed-Solomon decoder, there are a finite number of errors per Reed-Solomon data block that can be corrected. The larger the number of errors introduced through blurring, the smaller the number of genuine errors that can be corrected.

17.1.3 Locate the Position Tag

The captured image is a 96×96 pixel map of 8-bit samples. Before any position information can be decoded, the position tag must be located within the image. Note that the tag can have any orientation.

The first step in locating the tag is to locate a bull's-eye target (Figure 41) within the image. Since we are imaging a wider area than a single tag it is possible that two targets lie within the image area.

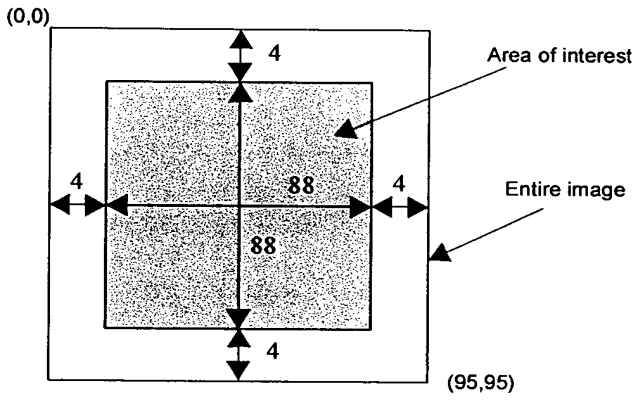


Figure 48. Area of interest for target location

Since an entire tag must be completely contained within the image, the data area outside the target must be visible. As a result, the area of interest for locating targets is actually a subset of the whole image. Instead of searching a 96×96 image, we only need search an 88×88 image (a reduction of 15% from 9216 samples to 7744).

The 5-layer bull's-eye pattern of the target is easily found in a 2D image by scans for topological connectivity, and is then useful for pinpointing the exact center regardless of orientation, and for determining the main axes and local x-y dimensions.

17.1.3.1 Scan for Topological Connectivity

Prior to image traversal we initialize an 88×88 array of flags with 0. This array keeps the visited status of the corresponding pixel in the image. If the flag is 0 we have not yet visited the pixel. If it is 1 or 2, then we have visited the pixel and bit 1 of the flag contains the sense of the bit (1 for black, 0 for white).

We then start with the top left pixel and compare adjacent pixels in order to build a connectivity structure for the top left pixel. If the top left pixel is considered to be white, then the connectivity structure will represent all connected pixels that are also white. If the top pixel is considered to be black, then the connectivity structure will represent all connected pixels that are also black. As edges are met due to differences in color sense, adjacent connectivity structures are built.

The order of pixel traversal in the image simply follows a standard pixel fill algorithm. There are many such algorithms to be found since this is a useful operation for paint programs. Near-optimal algorithms such as [32] and [36] only read each pixel once on average.

In our case we must also consider the notion of color sense. Target location does not require the sophistication of the macrodot reading algorithm for mode bits and data bits for determining if a macrodot is black or white (Section 17.1.4.2 and Section 17.1.4.4). The minimum and maximum pixels encountered during the image capture were stored in MinPixelEncountered and MaxPixelEncountered respectively, and these are used to set the threshold for black/white as MidRange = (MinPixelEncountered + MaxPixelEncountered)/2. Anything greater than or equal to MidRange is considered to be white, and anything less than MidRange is considered to be black.

As each connectivity structures is completed, it is compared against the expected values for the 5 bull's-eye layers surrounding the target center. Possible matches are added to one list, while non-matches are discarded.

The total time taken for processing the image is approximately 6 cycles per sample pixel. This equates to 46,464 cycles (88×88×6), or 19% of the available processor bandwidth.

17.1.3.2 Determine Target Axes

The connectivity structures can now be examined together in order to locate the target. The identification of two or more concentric structures is enough to determine the x and y axes.

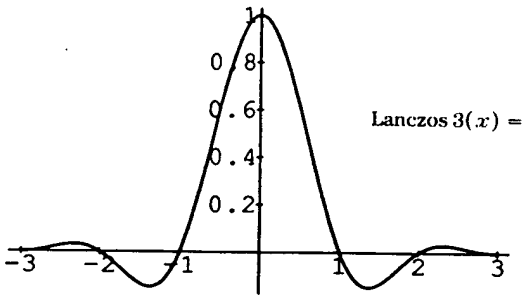
Taking one of the completely isolated concentric structures, 2 points on each of the 4 sides are chosen where the distance separating them is at least 3 pixels. For each point, the pixel of maximum whiteness or maximum blackness is chosen as the estimate of the center of the line. The estimate should be within 1 pixel of the actual center.

The process of building a more accurate position for the line center involves reconstructing the continuous signal for 7 scanline slices of the line, 3 to either side of the estimated center. The 7 maximum values found (one for each of these pixel dimension slices) are then used to reconstruct a continuous signal in the column dimension and thus to locate the maximum value in that dimension.

```
// Given estimates column and pixel, determine a
// betterColumn and betterPixel as the center of the target
for (y=0; y<7; y++)
{
   for (x=0; x<7; x++)
      samples[x] = GetPixel(column-3+y, pixel-3+x)
   FindMax(samples, pos, maxVal)
   reSamples[y] = maxVal
   if (y == 3)
      betterPixel = pos + pixel
```

```
}
FindMax(reSamples, pos, maxVal)
betterColumn = pos + column
```

FindMax is a function that reconstructs the original one-dimensional signal based sample points and returns the position of the maximum as well as the maximum value found. The method of signal reconstruction/resampling used is the Lanczos3 windowed sinc function (see pages 157-158 in [85]). The function and kernel are shown in Figure 49:



$$\text{Lanczos}\,3(x) = \begin{cases} \dfrac{\sin(\pi x)}{\pi x}\,\dfrac{\sin\left(\pi\frac{x}{3}\right)}{\pi\frac{x}{3}}, & |x| < 3 \\ 0, & |x| \geq 3 \end{cases}$$

Figure 49. Lanczos3 windowed sinc function

The Lanczos3 windowed sinc function takes 7 (pixel) samples from the dimension being reconstructed, centered around the estimated position x, i.e. at x-3, x-2, x-1, x, x+1, x+2, x+3. We reconstruct points from x-1 to x+1, each at an interval of 0.1, and determine which point is the maximum. The position that is the maximum value becomes the new center. Due to the nature of the kernel, we only require 6 entries in the convolution kernel for points between x and x+1. We use 6 points for x-1 to x, and 6 points for x to x+1, requiring 7 points overall in order to get pixel values from x-1 to x+1 since some of the pixels required are the same.

Two accurate positions on each of 2 sides of the isolated bull's-eye layer give the x and y axes directly. Using 4 sides allows an average line to be calculated (simply the average of the two lines built from opposite sides).

The distance from the center of one bull's-eye layer macrodot line to the center of the next bull's-eye layer's macrodot line gives the distance between macrodots in terms of a Δx and a Δy with respect to the current orientation of the image.

17.1.3.3 Determine Target Center

In order to locate the actual center of the target there are two cases to consider.

- The center of the target is one of the connectivity structures
- The center of the target is *not* one of the connectivity structures

In the first case, we choose the blackest pixel of the estimated target center, which should be within 1 pixel of the actual center. The process of building a more accurate position for the target center involves reconstructing the continuous signal for 7 scanline slices of the target, 3 to either side of the estimated target center. The 7 maximum values found (one for each of these pixel dimension slices) are then used to reconstruct a continuous signal in the column dimension and thus to locate the maximum value in that dimension. The process is the same as described in Section 17.1.3.2 and uses the same filter as shown in Figure 49.

If there is no current estimation for the center, the center is considered to be the calculated center of the four lines used in the calculation of the axes in Section 17.1.3.2. This process can be repeated for another isolated bull's-eye layer and the two estimates for target center averaged. If the estimated center pixel is black, a more accurate center can potentially be obtained by following the procedure defined for the first case. However care must be taken since this pixel region is not completely isolated (or there would have been a target estimate in the connectivity structures). It may be enough to increase the accuracy in only one dimension.

17.1.4 Decode the Position Tag

With the position of the tag known, the task of decoding it can begin. Decoding a positional tag consists of 3 essential steps:

- Determining the orientation of the tag
- Extracting the tag's mode data and decoding it
- Extracting and decoding the data portion of the tag based on the mode data

17.1.4.1 Determine Orientation

On the four corners of the target are 3-bit clusters of orientation bits, which allow the tag's orientation to be quickly determined.

Using the Δx and Δy values (obtained in Section 17.1.3.2), and applying them to the accurate target center coordinate (obtained in Section 17.1.3.3), we are able to calculate the estimated positions of the 12 orientation macrodots.

Given the macrodot coordinate (fixed point) we sample 4 image pixels to arrive at a center pixel value via bilinear interpolation.

Once the center pixel value has been determined, we try to determine the bit value for that macrodot. To do so, we take the pixel values representing the macrodot centers to either side of the macrodot whose bit value is being determined, and attempt to intelligently guess the value of that center macrodot's bit value. Looking at the generalized blurring curve again (reproduced in Figure 50 from Figure 46), but with BlackMax and WhiteMin shown, there are three common cases to consider:

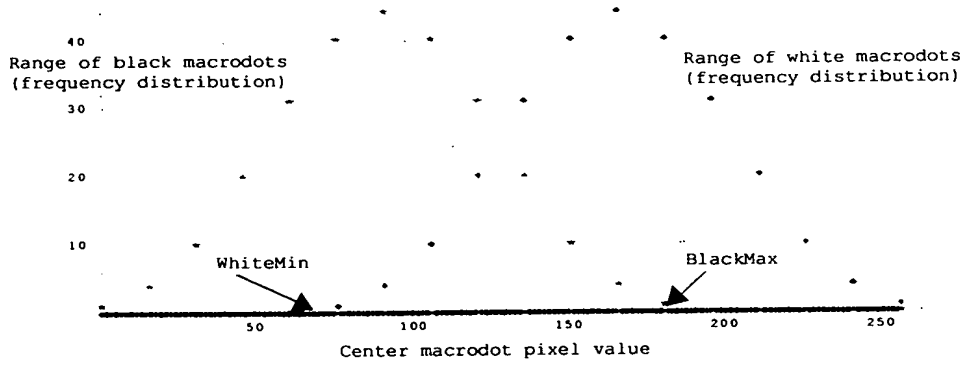


Figure 50. Black-white macrodot ambiguity due to blurring

- The macrodot's center pixel value is lower than WhiteMin, and is therefore definitely a black macrodot. The bit value is therefore definitely 1.

- The macrodot's center pixel value is higher than BlackMax, and is therefore definitely a white macrodot. The bit value is therefore definitely 0.
- The macrodot's center pixel value is somewhere between BlackMax and WhiteMin. The macrodot may be black, and it may be white. The value for the bit is therefore in question. A number of schemes can be devised to make a reasonable guess as to the value of the bit. These schemes must balance complexity against accuracy, and also take into account the fact that in some cases, there is no guaranteed solution. In those cases where we make a wrong bit decision, the bit will be in error, and must be corrected by some other means (in the case of orientation bits, the other corners may help, and in the case of the tag's data, the Reed-Solomon decoding stage will correct errors).

Before the data can be extracted from the data area, the pixel ranges for "black" and white dots needs to be ascertained. The minimum and maximum pixels encountered during the search for the tag were stored in MinPixelEncountered and MaxPixelEncountered respectively. The following pseudocode shows the method of obtaining good values for WhiteMin and BlackMax based on the min & max pixels encountered.

```
MinPixel = MinPixelEncountered
MaxPixel = MaxPixelEncountered
MidRange = (MinPixel + MaxPixel) / 2
WhiteMin = MaxPixel - PenConstantWhiteFactor
BlackMax = MinPixel + PenConstantBlackFactor
```

The scheme used to determine a macrodot's value if the pixel value is between BlackMax and WhiteMin is fairly simple, but gives good results. It uses the pixel values of the macrodot centers to the left and right of the dot in question, using their values to help determine a more likely value for the center dot:

- If the two macrodots to either side are on the white side of MidRange (an average macrodot value), then we can guess that if the center macrodot were white, it would likely be a "definite" white. The fact that it is in the not-sure region would indicate that the macrodot was black, and had been affected by the surrounding white macrodots to make the value less sure. The macrodot value is therefore assumed to be black, and hence the bit value is 1.
- If the two macrodots to either side are on the black side of MidRange, then we can guess that if the center macrodot were black, it would likely be a "definite" black. The fact that it is in the not-sure region would indicate that the macrodot was white, and had been affected by the surrounding black macrodots to make the value less sure. The macrodot value is therefore assumed to be white, and hence the bit value is 0.
- If one macrodot is on the black side of MidRange, and the other macrodot is on the white side of MidRange, we simply use the center macrodot value to decide. If the center macrodot is on the black side of MidRange, we choose black (bit value 1). Otherwise we choose white (bit value 0).

The logic is represented by this simple pseudocode:

```
if (pixel < WhiteMin)          // definitely black
   bit = 0x01
else
if (pixel > BlackMax)          // definitely white
   bit = 0x00
else
```

```
if ((prev > MidRange) && (next> MidRange))     //prob black
   bit = 0x01
else
if ((prev < MidRange) && (next < MidRange))    //prob white
   bit = 0x00
else
if (pixel < MidRange)
   bit = 0x01
else
   bit = 0x00
```

From this one can see that using surrounding pixel values can give a good indication of the value of the center macrodot's state. The scheme described here only uses the macrodots from the same row, but increased accuracy may be achieved using all the neighboring macrodots, at the cost of increased complexity.

We already have the distance between the macrodots in terms of a Δx and a Δy with respect to the current orientation of the image. Rather than rotate the image to transform Δx and Δy both to 1, we simply change the sense of the Δx and Δy values based on how many sets of 90 degrees it takes for the tag to be oriented so that both D values are positive. The four possible rotations by 90 degrees equate to 4 different combinations of signs ($\pm\Delta x$ and $\pm\Delta y$). It is therefore trivial to translate Δx and Δy into two deltas for movement in x and y: Δx_x and Δy_x for movement of 1 macrodot in x, and Δx_y and Δy_y for movement of 1 macrodot in y.

With the 12 orientation bits obtained, the orientation of the tag can be easily determined. Given 4 sets of 12 bits, the correct orientation is the one that minimizes the number of bits in error, as shown in Table 8.

Table 8. The four major orientations of the position tag

12-bit value	Orientation of tag
111-011-100-000	No rotation
011-100-000-111	Rotated clockwise 90 degrees
100-000-111-011	Rotated clockwise 180 degrees
000-111-011-100	Rotated clockwise 270 degrees

17.1.4.2 Extract the Tag's Mode Raw Data

With the orientation determined, the next step is to read the bit pattern representing the Mode bits from the tag.

We know the coordinate of the tag's target center as $[Center_x, Center_y]$. We also know the orientation of the tag and therefore have Δx_x and Δy_x for motion in x, and Δx_y and Δy_y for movement in y. These delta values enable us to step along the center of any macrodots. By sampling the tag at the 18 mode bit positions (see Figure 42). Consequently the location of the first few mode bit are:

- Bit0 location = $[Center_x - 5\Delta x_x - 5\Delta x_y, Center_y - 5\Delta y_x - 5\Delta y_y]$
- Bit1 location = $[Bit0_x + \Delta x_x, Bit0_y + \Delta y_x]$
- Bit2 location = $[Bit1_x + \Delta x_x, Bit1_y + \Delta y_x]$

Given the macrodot coordinate (fixed point) we use the sampling mechanism as described in Section 17.1.4.1 to extract the bit value.

The extraction of 28 bits will take approximately 168 cycles on a simple microprocessor. Assuming a clock speed of 25 MHz, 168 cycles per position sample represents less than 0.1% of the available processor bandwidth.

17.1.4.3 Decode the Mode Data

The 28-bit raw bitstream representing the tag's mode data has been read in. It must now be decoded into 8 bits of meaningful mode information.

The data is decoded using straightforward Reed-Solomon decoding. One of the advantages of using Galois Field GF(16) is that a total of only 64 bytes are required for lookup tables during the decoding process (16 bytes each of power, log, multiply and multiply inverse).

The decoding process of 28 bits to 8 bits will take approximately 1500 cycles on a simple microprocessor. Assuming a clock speed of 25 MHz, 1500 cycles per position sample is less than 1% of the available processor bandwidth.

17.1.4.4 Extract the Raw Tag Data

Before the tag data *proper* can be read in from the tag image, the tag's 8-bit mode data must be examined. The Version bit must be checked to ensure that this tag has a known structure (see Table 6). If the Version bit is 0, then the tag data structure is known and can be read in from the image.

The macrodots representing the tag data area are read into a 168-bit raw bitstream according to the locations described in Figure 43. We use the tag's target center [Center_x, Center_y] and the orientation delta values Δx_x and Δy_x, Δx_y and Δy_y to calculate the start position. The delta values also enable us to step along the center of the data area's macrodots.

The process of determining if a given macrodot is a 1 or a 0 is the same as that carried out for reading the tag's mode bits and orientation bits (see Section 17.1.4.1). A total of 168 bits is read in from the image in this way. The extraction of 168 bits will take approximately 1008 cycles on a simple microprocessor. Assuming a clock speed of 25 MHz, 1008 cycles per position sample represents 0.4% of the available processor bandwidth.

17.1.4.5 Decode the Tag Data

The 168-bit raw bitstream representing the tag data has been read in. It must now be decoded into the PageId, x and y coordinates etc. (see Table 5).

The 28 × 6-bit codewords can be divided into message data codewords and check words. Assuming that the Version bit is 0, the 2-bit NumMsgWords field in the Mode area determines the number of message data codewords D as 15, 16, 17, or 18. The remaining K codewords are check codewords.

Each of the D 6-bit message codewords can now be examined to see if it is all 0s or all 1s. Any such codeword is illegal (due to the encoding process) and can be considered to have been erased. Determining that a codeword has been erased increases the error correction capabilities of the check bits.

The data can then be decoded using straightforward Reed-Solomon decoding. One of the advantages of using Galois Field GF(64) is that a total of only 256 bytes are required for

lookup tables during the decoding process (64 bytes each of power, log, multiply and multiply inverse).

The decoding process of 168 bits to 90 bits will take approximately 4000 cycles on a simple microprocessor. Assuming a clock speed of 25 MHz, 4000 cycles per position sample represents 1.6% of the available processor bandwidth.

17.1.5 Build Stroke Information

The information gained from the position tag is either ignored or added to the current stroke depending on whether the pen is deemed, based on nib pressure, to be up or down. If the pen is up but was down, then the current stroke is finished. If the pen was up but is now down, then a new stroke is started.

Each stroke contains the start time (in milliseconds) followed by the recovered page id and initial position. This is followed by a series of positions for the stroke. The positions are implicitly separated by 100ths of a second in time. While the first position is always absolute, subsequent positions are either delta-encoded or absolute as required. Escape codes allow compression for a number of unknown positions (the tag cannot be found), the encounter of different page ids (for example the initial page id is unknown and finally a page id is recovered from a tag during the stroke, or it may be that the pen crosses from one page to another during a stroke), and the encountering of tags with differing PressureSensitiveArea values, which enable or disable the inclusion of a pressure value with each position. Pressure values are typically included for the entire stroke if the stroke begins in a pressure-sensitive area.

The following is a definition for a stroke:

```
Stroke = time PageIdRec FirstPosRec {[PosRec | PressurePos | Escape]} endStroke.
PageIdRec = [unknown | pageId]
Escape = [pageId | pressureMode | TimeSkip]
TimeSkip = [unknown | zeroDelta] shortTime
PressurePos = PosRec penPressure
PosRec = [PenAbsRec | PenDeltaRec]
FirstPosRec = [unknown | PenAbsRec]
PenAbsRec = tagCoordinate imageCoordinate macrodotDelta tilt
PenDeltaRec = penDelta imageCoordinate macrodotDelta tilt

where:
time = 32 bits unsigned (milliseconds)
shortTime = 16 bits unsigned (milliseconds)
pageId = 64 bits unsigned
tagCoordinate = 2 x 9 bits unsigned (representing x and y)
imageCoordinate = 2 x 10 bits unsigned fixed point 7:3 (representing x and y)
macrodotDelta = 2 x 8 bits signed fixed point S3:4 (representing Δx and Δy)
penDelta = 2 x 4 bits signed (representing -8..7 for x and y)
penPressure = 5 bits
tilt = 2 x 5 bits (representing 2 tilt axes)
pressureMode = 1 bit
```

The length of a sample position is 64 bits (18 + 20 + 16 + 10) when the pen coordinate is fully described, and 52 bits (18 + 8 + 16 + 10) when the pen coordinate is a delta amount. Adding flag bits to specify that there is no escape mode (1 bit) and to select between absolute and relative position (1 bit), we specify 66 and 54 bits respectively.

The initial position requires 162 bits to define (32 + 65 + 65).

For a stroke over a single page that lasts 4 seconds, and assuming no read errors, there is an initial pen down followed by 399 deltas. The total stroke length is therefore 21,708 bits (162 + (399 × 54)), which is 2714 bytes, or 6.8 bytes per sample.

If pressure is also captured with each position (depending on the value of the encountered PressureSensitiveArea bits within each tag), a further 5 bits are required for each position. The Escape of pressureMode allows the toggling of this state within a stroke.

The stroke definition copes well with unknown pen locations. If the initial position (pageId, coordinates etc.) is unknown due to the inability to locate a tag or recover tag data, the initial position for the stroke is marked as unknown. This initial position may be followed by an Escape of TimeSkip, signifying the amount of time where unknown positions were captured. Finally, when a tag is captured and decoded successfully, the Escape for pageId is used. This pageId is valid until the next Escape pageId is included in the stroke (inserted when a tag is encountered with a pageId different to the current *known* pageId). If a tag cannot be found or decoded successfully mid-stream within a stroke, the Escape for TimeSkip is used, which represents the elapsed time where only unknown positions were sampled. An unknown pen position does not change the pageId. If tag positions are reacquired, the page id may be changed, depending on the tag. A Timeskip entry for a set of unknown positions uses 1 bit for the escape code, 2 bits to define the type of escape code, and 16 bits to define the duration of samples skipped.

The other use for the Escape of TimeSkip is when there have been one or more deltas of 0. The Escape of zeroDelta combines with a 16-bit time amount to specify how long the pen has remained in the same position without moving.

The pen contains 0.5MB of RAM to buffer stroke information both during stroke construction and if the pen is out of range of the printer. A 0.5MB buffer allows for approximately 192 strokes at 4 seconds per stroke, which represents more than 12 minutes of stroke capture. The exact duration will depend on the success of tag decoding, and whether or not continuous pressure is being captured.

If the stroke buffer becomes full, a *Buffer-Full* LED is illuminated. The LED stays illuminated until the pen comes into range of a printer, whereupon the buffered strokes are transmitted to the printer and the buffer freed.

17.1.6 Activate Active Area Light

The position tag data area contains a ActiveArea bit (see Table 5). The pen's active area LED must be enabled directly in response to the value of this bit. There is no additional interpretation of the ActiveArea bit by the pen.

17.2 DATA ENCRYPTION

The data stream must be encrypted before being transmitted to the printer, to prevent eavesdroppers from capturing pen input, and particularly signature input.

Consequently, the final stage before transmission is the encryption of the data stream. Symmetric encryption is used since it is tractable in the time available. Possible symmetric

algorithms for use are listed in Table 9. In all cases, a minimum key size of 14 bytes is assumed.

Table 9. Symmetric encryption algorithms [72]

Algorithm	Type	Clocks per byte processed (Pentium)	Block size (bytes)	Key size (bytes)	Patents
Blowfish	Block	18	8	4-56	No
CAST	Block	20	8	16	Yes, but free
IDEA	Block	50	8	16	Yes
Triple-DES	Block	108	8	14[a]	No

a. Although Triple-DES uses a 168 bit key, the effective key strength is only 112 bits

Of the algorithms listed in Table 9, Triple-DES is the most conservative choice, although it is the slowest. At 108 clocks per byte processed, the encoding of a single 8-byte block consumes 864 cycles (8 × 108).

The pen must send 100 encrypted samples each second. Assuming that each sample requires a complete Triple-DES block (8 bytes), the time taken to encrypt 1 second of data is 86,400 cycles.

For a clock speed of 25 MHz, 86,400 cycles for the 100 samples equates to less than 0.5% of the processor bandwidth.

The encryption process therefore has a throughput of 800 bytes per second.

PRINT ENGINE

18 Print Engine Page Format

This section describes the format of compressed pages expected by the print engines.

The raster image processors (RIPs) generate pages in this format. The compressed format and the print engines are designed to allow real-time page expansion during printing, to ensure that printing is never interrupted in the middle of a page due to data underrun.

The Netpage Printer reproduces black at full dot resolution (1600 dpi), but reproduces contone color at a somewhat lower resolution using halftoning. The page description is therefore divided into a black layer and a contone layer. The black layer is defined to composite *over* the contone layer.

The black layer consists of a bitmap containing a 1-bit *opacity* for each pixel. This black layer *matte* has a resolution which is an integer factor of the printer's dot resolution. The highest supported resolution is 1600 dpi, i.e. the printer's full dot resolution.

The contone layer consists of a bitmap containing a 24-bit CMY *color* for each pixel. This contone image has a resolution which is an integer factor of the printer's dot resolution. The highest supported resolution is 267 ppi, i.e. one-sixth the printer's dot resolution.

The contone resolution is also typically an integer factor of the black resolution, to simplify calculations in the RIPs. This is not a requirement, however.

The black layer and the contone layer are both in compressed form for efficient storage in the printer's internal memory.

18.1 Page Structure

The Netpage Printer prints with full edge bleed using an 8½" printhead. It imposes no margins and so has a printable page area which corresponds to the size of its paper. The target page size is constrained by the printable page area, less the explicit (target) left and top margins specified in the page description. These relationships are illustrated below.

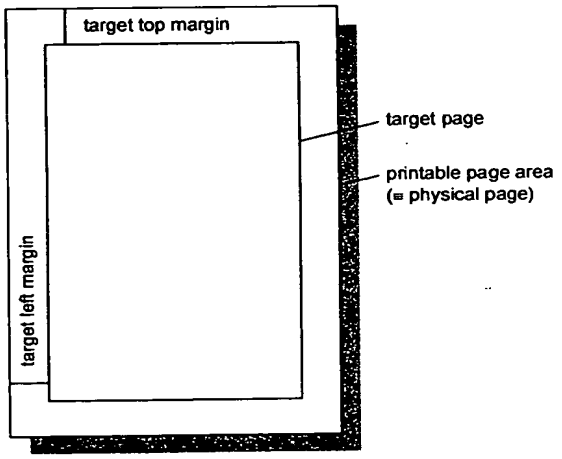


Figure 51. Page structure

18.2 COMPRESSED PAGE FORMAT

Apart from being implicitly defined in relation to the printable page area, each page description is complete and self-contained. There is no data stored separately from the page description to which the page description refers.

The page description consists of a page header which describes the size and resolution of the page, followed by one or more page bands which describe the actual page content.

Table 10 shows the format of the page header.

Table 10. Page header format

field	format	description
signature	16-bit integer	Page header format signature.
version	16-bit integer	Page header format version number.
structure size	16-bit integer	Size of page header.
target resolution (dpi)	16-bit integer	Resolution of target page. This is always 1600 for the Netpage Printer.
target page width	16-bit integer	Width of target page, in dots.
target page height	16-bit integer	Height of target page, in dots.
target left margin	16-bit integer	Width of target left margin, in dots.
target top margin	16-bit integer	Height of target top margin, in dots.
black scale factor	16-bit integer	Scale factor from black resolution to target resolution (must be 2 or greater).
black page width	16-bit integer	Width of black page, in black pixels.
black page height	16-bit integer	Height of black page, in black pixels.
contone scale factor	16-bit integer	Scale factor from contone resolution to target resolution (must be 6 or greater).
contone page width	16-bit integer	Width of contone page, in contone pixels.
contone page height	16-bit integer	Height of contone page, in contone pixels.

The page header contains a signature and version which allow the print engine to identify the page header format. If the signature and/or version are missing or incompatible with the print engine, then the print engine can reject the page.

The page header defines the resolution and size of the target page. The black and contone layers are clipped to the target page if necessary. This happens whenever the black or contone scale factors are not factors of the target page width or height.

The target left and top margins define the positioning of the target page within the printable page area.

The black layer parameters define the pixel size of the black layer, and its integer scale factor to the target resolution.

The contone layer parameters define the pixel size of the contone layer, and its integer scale factor to the target resolution.

Table 11 shows the format of the page band header.

Table 11. Page band header format

field	format	description
signature	16-bit integer	Page band header format signature.
version	16-bit integer	Page band header format version number.
structure size	16-bit integer	Size of page band header.
black band height	16-bit integer	Height of black band, in black pixels.
black band data size	32-bit integer	Size of black band data, in bytes.
contone band height	16-bit integer	Height of contone band, in contone pixels.
contone band data size	32-bit integer	Size of contone band data, in bytes.
watermark map band data size	32-bit integer	Size of watermark map band data, in bytes.
tag control band height	16-bit integer	Height of position tag control band, in tags.
active area map band size	32-bit integer	Size of active area map band data, in bytes.
pressure-sensitive area map band size	32-bit integer	Size of pressure-sensitive area band data, in bytes.

The black layer parameters define the height of the black band, and the size of its compressed band data. The variable-size black data follows the page band header.

The contone layer parameters define the height of the contone band, and the size of its compressed page data, consisting of the contone color data and the associated bi-level watermark map. The variable-size contone data follows the black data. The variable-size bi-level watermark map data follows the contone data.

The tag layer parameters define the height of the position tag control band, and the size of its active area and pressure-sensitive area map band data. The variable-size active area map data follows the watermark map data. The variable-size pressure-sensitive area map data follows the active area map data.

Table 12 shows the format of the variable-size compressed band data which follows the page band header.

Table 12. Page band data format

field	format	description
black data	G4Fax bytestream	Compressed bi-level black data.
contone data	JPEG bytestream	Compressed contone CMY data.
watermark map	G4Fax bytestream	Compressed bi-level watermark map data.
active area map	bitmap	Bi-level active area map data.
pressure-sensitive area map	bitmap	Bi-level pressure-sensitive area map data.

Each variable-size segment of band data is aligned to an 8-byte boundary.

The following sections describe the format of the compressed bi-level layers and the compressed contone layer.

18.2.1 Bi-level Data Compression

18.2.1.1 Group 3 and 4 Facsimile Compression

The Group 3 Facsimile compression algorithm [10] losslessly compresses bi-level data for transmission over slow and noisy telephone lines. The bi-level data represents scanned black text and graphics on a white background, and the algorithm is tuned for this class of images (it is explicitly not tuned, for example, for *halftoned* bi-level images). The *1D* Group 3 algorithm runlength-encodes each scanline and then Huffman-encodes the resulting runlengths. Runlengths in the range 0 to 63 are coded with *terminating* codes. Runlengths in the range 64 to 2623 are coded with *make-up* codes, each representing a multiple of 64, followed by a terminating code. Runlengths exceeding 2623 are coded with multiple make-up codes followed by a terminating code. The Huffman tables are fixed, but are separately tuned for black and white runs (except for make-up codes above 1728, which are common). When possible, the *2D* Group 3 algorithm encodes a scanline as a set of short edge deltas (0, ±1, ±2, ±3) with reference to the previous scanline. The delta symbols are entropy-encoded (so that the zero delta symbol is only one bit long etc.) Edges within a 2D-encoded line which can't be delta-encoded are runlength-encoded, and are identified by a prefix. 1D- and 2D-encoded lines are marked differently. 1D-encoded lines are generated at regular intervals, whether actually required or not, to ensure that the decoder can recover from line noise with minimal image degradation. 2D Group 3 achieves compression ratios of up to 6:1 [87].

The Group 4 Facsimile algorithm [10] losslessly compresses bi-level data for transmission over *error-free* communications lines (i.e. the lines are truly error-free, or error-correction is done at a lower protocol level). The Group 4 algorithm is based on the 2D Group 3 algorithm, with the essential modification that since transmission is assumed to be error-free, 1D-encoded lines are no longer generated at regular intervals as an aid to error-recovery. Group 4 achieves compression ratios ranging from 20:1 to 60:1 for the CCITT set of test images [87].

The design goals and performance of the Group 4 compression algorithm qualify it as a compression algorithm for the bi-level layers. However, its Huffman tables are tuned to a lower scanning resolution (100-400 dpi), and it encodes runlengths exceeding 2623 awkwardly. At 800 dpi, our maximum runlength is currently 6400. Although a Group 4 decoder core might be available for use in the printer controller chip, it might not handle runlengths exceeding those normally encountered in 400 dpi facsimile applications, and so would require modification.

18.2.1.2 Edge Delta / Runlength (EDRL) Compression

Since most of the benefit of Group 4 comes from the delta-encoding, a simpler algorithm based on delta-encoding alone is likely to meet our requirements. This approach, in the form of the Edge Delta / Runlength (EDRL) encoding, is described in detail in another Silverbrook design document [3]. There are therefore two viable approaches to losslessly compressing bi-level data: Group 4 Facsimile and EDRL. Their compression performance is compared below. Where this document refers to Group 4 Facsimile (G4Fax) compression, EDRL can be substituted, purely as an implemetation decision.

Magazine text is typically typeset in a typeface with serifs (such as Times) at a point size of 10. At this size a Letter page holds up to 14,000 characters, though a typical magazine page holds only about 7,000 characters. Text is seldom typeset at a point size smaller than 5. At 800 dpi, text cannot be meaningfully rendered at a point size lower than 2 using a standard typeface. Table 13 illustrates the legibility of various point sizes.

Table 13. Text at different point sizes

point size	sample text (in Times)
2	
3	
4	
5	The quick brown fox jumps over the lazy dog.
6	The quick brown fox jumps over the lazy dog.
7	The quick brown fox jumps over the lazy dog.
8	The quick brown fox jumps over the lazy dog.
9	The quick brown fox jumps over the lazy dog.
10	The quick brown fox jumps over the lazy dog.

Table 14 shows Group 4 and EDRL compression performance on pages of text of varying point sizes, rendered at 800 dpi. The distribution of characters on the test pages is based on English-language statistics [84].

Table 14. Group 4 and EDRL compression performance on text at 800 dpi

point size	characters per page	Group 4 compression ratio	EDRL compression ratio
2	340,000	2.3	1.7
3	170,000	3.2	2.5
4	86,000	4.7	3.8
5	59,000	5.5	4.9
6	41,000	6.5	6.1
7	28,000	7.7	7.4
8	21,000	9.1	9.0
9	17,000	10.2	10.4
10	14,000	10.9	11.3
11	12,000	11.5	12.4
12	8,900	13.5	14.8
13	8,200	13.5	15.0
14	7,000	14.6	16.6
15	5,800	16.1	18.5
20	3,400	19.8	23.9

For a point size of 9 or greater, EDRL slightly outperforms Group 4, simply because Group 4's runlength codes are tuned to 400 dpi.

These compression results bear out the observation that entropy-encoded runlengths contribute much less to compression than 2D encoding, unless the data is poorly correlated vertically, such as in the case of very small characters.

18.2.1.3 Black Layer Compression Format

The 800 dpi black layer is losslessly compressed using G4Fax (or EDRL) at a typical compression ratio exceeding 10:1.

18.2.1.4 Watermark Map Compression Format

The 267 dpi watermark layer, which matches the 267 ppi contone color layer, is losslessly compressed using G4Fax (or EDRL) at a typical compression ratio exceeding 50:1.

18.2.2 Contone Data Compression

18.2.2.1 JPEG Compression

The JPEG compression algorithm [45] lossily compresses a contone image at a specified quality level. It introduces imperceptible image degradation at compression ratios below 5:1, and negligible image degradation at compression ratios below 10:1 [88].

JPEG typically first transforms the image into a color space which separates luminance and chrominance into separate color channels. This allows the chrominance channels to be subsampled without appreciable loss because of the human visual system's relatively greater sensitivity to luminance than chrominance. After this first step, each color channel is compressed separately.

The image is divided into 8×8 pixel blocks. Each block is then transformed into the frequency domain via a discrete cosine transform (DCT). This transformation has the effect of concentrating image energy in relatively lower-frequency coefficients, which allows higher-frequency coefficients to be more crudely quantized. This quantization is the principal source of compression in JPEG. Further compression is achieved by ordering coefficients by frequency to maximize the likelihood of adjacent zero coefficients, and then runlength-encoding runs of zeroes. Finally, the runlengths and non-zero frequency coefficients are entropy coded. Decompression is the inverse process of compression.

18.2.2.2 CMY Contone Layer Compression Format

The CMY contone layer is compressed to an interleaved color JPEG bytestream. The interleaving is required for space-efficient decompression in the printer, but may restrict the decoder to two sets of Huffman tables rather than four (i.e. one per color channel) [88]. If luminance and chrominance are separated, then the luminance channels can share one set of tables, and the chrominance channels the other set.

If luminance/chrominance separation is deemed necessary, either for the purposes of table sharing or for chrominance subsampling, then CMY is converted to YCrCb and Cr and Cb are duly subsampled.

The JPEG bytestream is complete and self-contained. It contains all data required for decompression, including quantization and Huffman tables.

19 Print Engine

19.1 HALFTONER/COMPOSITOR

The halftoner/compositor unit (HCU) (Figure 53) combines the functions of halftoning the contone CMY layer to bi-level CMY, and compositing the black layer over the halftoned contone layer. It also selects between the normal and the "watermark" dither matrix on a pixel-by-pixel basis, based on the corresponding value in the watermark map.

The input to the HCU is an expanded 267 ppi CMY contone layer, an expanded 267 dpi watermark map, and an expanded 1600 dpi black layer. The output from the HCU is a set of 1600 dpi bi-level CMY image lines.

Once started, the HCU proceeds until it detects an *end-of-page* condition, or until it is explicitly stopped via its control register.

The HCU generates a page of dots of a specified width and length. The width and length must be written to the *page width* and *page length* registers prior to starting the HCU. The page width corresponds to the width of the printhead. The page length corresponds to the length of the target page.

The HCU generates target page data between specified left and right margins relative to the page width. The positions of the left and right margins must be written to the *left margin* and *right margin* registers prior to starting the HCU. The distance from the left margin to the right margin corresponds to the target page width.

The HCU consumes black and contone/watermark data according to specified black and contone page widths. These page widths must be written to the *black page width* and *contone page width* registers prior to starting the HCU. The HCU clips black and contone data to the target page width. This allows the black and contone page widths to exceed the target page width without requiring any special end-of-line logic at the input FIFO level.

The relationships between the page width, the black and contone page widths, and the margins are illustrated in Figure 52.

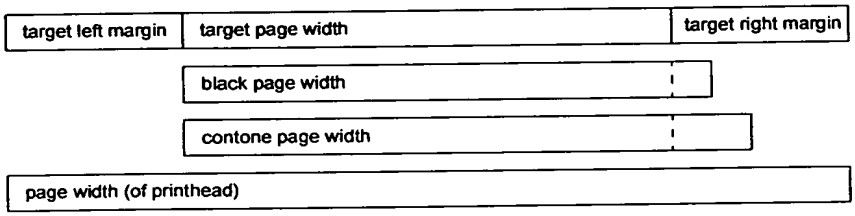


Figure 52. Relationships between page widths and margins

The HCU scales contone data to printer resolution both horizontally and vertically based on a specified scale factor. This scale factor must be written to the *contone scale factor* register prior to starting the HCU.

Table 15. Halftoner/compositor control and configuration registers

register	width	description
start	1	Start the HCU.
stop	1	Stop the HCU.
page width	14	Page width of printed page, in dots. This is the number of dots which have to be generated for each line.
left margin	14	Position of left margin, in dots.
right margin	14	Position of right margin, in dots.
page length	15	Page length of printed page, in dots. This is the number of lines which have to be generated for each page.
black page width	14	Page width of black layer, in dots. Used to detect the end of a black line.
contone page width	14	Page width of contone layer, in dots. Used to detect the end of a contone line.
contone scale factor	4	Scale factor used to scale contone data to bi-level resolution.

The consumer of the data produced by the HCU is the printhead interface. The printhead interface requires bi-level CMY image data in *planar* format, i.e. with the color planes separated. Further, it also requires that even and odd pixels are separated. The output stage of the HCU therefore uses 6 parallel pixel FIFOs, one each for *even cyan*, *odd cyan*, *even magenta*, *odd magenta*, *even yellow*, and *odd yellow*.

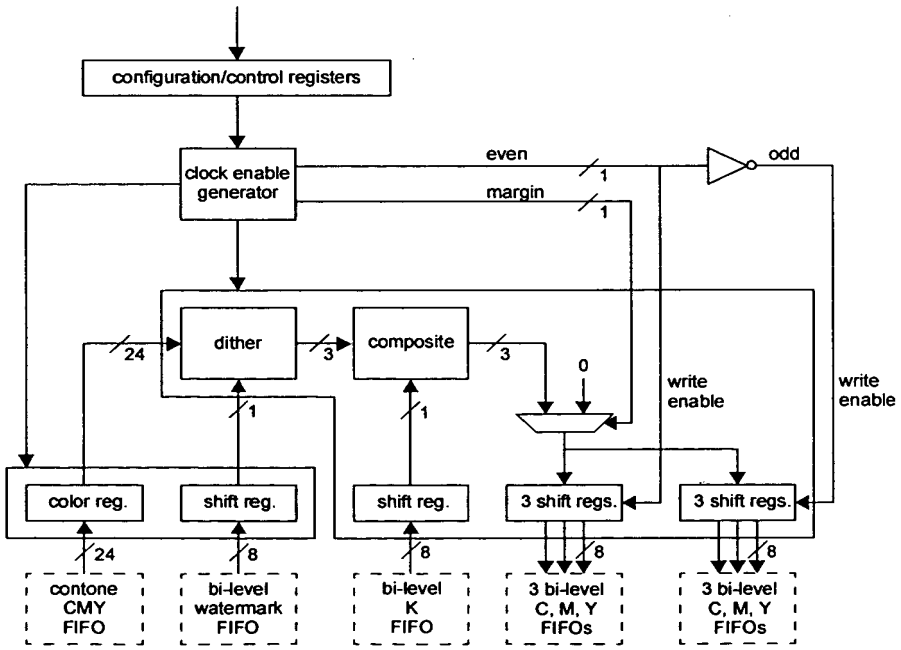


Figure 53. Halftoner/compositor unit

The input contone CMY FIFO is a full 7KB line buffer. The line is used *contone scale factor* times to effect vertical up-scaling via line replication. FIFO write address wrapping is disabled until the start of the last use of the line.

19.1.1 Dither

The dither unit converts a 24-bit contone CMY value into a 3-bit bi-level CMY value. Each color component is independently compared with the 8-bit threshold value at the current dot position in the dither cell (see Figure 54).

There are two programmable 64×64×8-bit dither cells: a normal dither cell and a "watermark" dither cell. The 1-bit watermark value associated with the current contone pixel controls the selection of the watermark or normal threshold value.

The dither cell address generator contains a pair of 6-bit row and column address registers. Both registers are initialized to zero. The row address is incremented on every dot line advance, modulo 64. The column address is incremented on every dot clock, modulo 64, and is reset on a dot line advance.

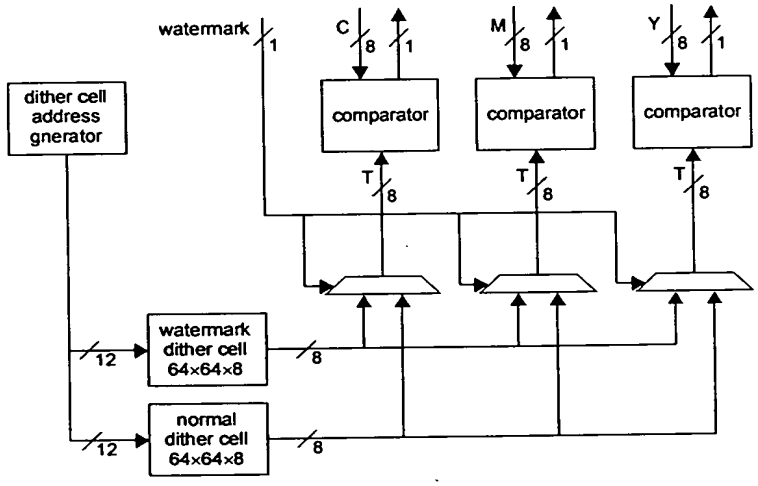


Figure 54. Dither, with watermarking

19.1.2 Composite

The composite unit composites a black layer dot over a halftoned CMY layer dot. If the black layer opacity is one, then the halftoned CMY is set to black (i.e. all ones). The black layer opacity is therefore simply ORed with each of the halftoned C, M, and Y values.

19.1.3 Clock Enable Generator

The clock enable generator generates enable signals for clocking the contone CMY/watermark pixel input, the black dot input, and the CMY dot output.

As described earlier, the contone pixel input buffer is used as both a line buffer and a FIFO. Each line is read once and then used *contone scale factor* times. FIFO write address wrapping is disabled until the start of the final replicated use of the line, at which time the clock enable generator generates a *contone line advance enable* signal which enables wrapping.

The clock enable generator also generates an *even* signal which is used to select the even or odd set of output dot FIFOs, and a *margin* signal which is used to generate white dots when the current dot position is in the left or right margin of the page.

The clock enable generator uses a set of counters. The internal logic of the counters is defined in Table 16. The logic of the clock enable signals is defined in Table 17.

Table 16. Clock enable generator counter logic

counter	abbr.	w.	data	load condition	decrement condition
dot	D	14	page width	RP[a] ∨ EOL[b]	(D>0) ∧ clk
line	L	15	page length	RP	(L>0) ∧ EOL
left margin	LM	14	left margin	RP ∨ EOL	(LM>0) ∧ clk
right margin	RM	14	right margin	RP ∨ EOL	(RM>0) ∧ clk
even/odd dot	E	1	0	RP ∨ EOL	clk
black dot	BD	14	black width	RP ∨ EOL	(LM=0) ∧ (BD>0) ∧ clk
contone dot	CD	14	contone width	RP ∨ EOL	(LM=0) ∧ (CD>0) ∧ clk
contone sub-pixel	CSP	4	contone scale factor	RP ∨ EOL ∨ (CSP=0)	(LM=0) ∧ clk
contone sub-line	CSL	4	contone scale factor	RP ∨ (CSL=0)	EOL ∧ clk

a. RP (reset page) condition: external signal
b. EOL (end-of-line) condition: (D=0) ∧ (BD=0) ∧ (CD=0)

Table 17. Clock enable generator output signal logic

output signal	condition
output dot clock enable	(D>0) ∧ ¬EOP[a]
black dot clock enable	(LM=0) ∧ (BD>0) ∧ ¬EOP
contone pixel clock enable	(LM=0) ∧ (CD>0) ∧ (CSP=0) ∧ ¬EOP
contone line advance enable	(CSL=0) ∧ ¬EOP
even	E=0
margin	(LM=0) ∨ (RM=0)

a. EOP (end-of-page) condition: L=0

19.2 PRINTHEAD INTERFACE

Netpage uses an 8½" CMYK Memjet printhead, as described in Section 21. The printhead consists of 17 segments arranged in 2 segment groups. The first segment group contains 9 segments, and the second group contains 8 segments. There are 13,600 nozzles of each color in the printhead, making a total of 54,400 nozzles.

The printhead interface is a standard Memjet printhead interface, as described in Section 22, configured with the following operating parameters:

- MaxColors = 4
- SegmentsPerXfer = 9
- SegmentGroups = 2

Although the printhead interface has a number of external connections, not all used for an 8½" printhead, so not all are connected to external pins on the print engine. Specifically, the value for SegmentGroups implies that there are only 2 SRClock pins and 2 SenseSegSelect pins. All 36 ColorData pins are required, however.

19.3 Position Tag Encoder

The position tag encoder encodes the page id of the page being printed, together with the current x-y position on the page, into an error-correctably encoded position tag which is subsequently printed in infrared (IR) ink on the page.

The position tag encoder takes the following as input:

- the page id
- a pressure sensitive area bitmap at tag resolution
- an active area bitmap at tag resolution

It writes a bi-level IR bitstream to the bi-level IR FIFO. The position tag encoder design as defined here allows a single page id per printed page, and provides for a page size of up to 21.7 inches per dimension (17 × 4 × 512 / 1600). In addition, the position tag encoder allows for both landscape and portrait orientations.

The position tag encoder consists of two stages connected by a buffer. The first stage, the *tag data encoder*, Reed-Solomon encodes the data to be placed in the tags. The second stage, the *tag formatter*, places the encoded data into the tag format and passes it on to the bi-level IR FIFO. Since the tag formatter runs one line of tags behind the tag data encoder, the tag data buffer is in fact a double-buffer. The relationship is shown in Figure 55.

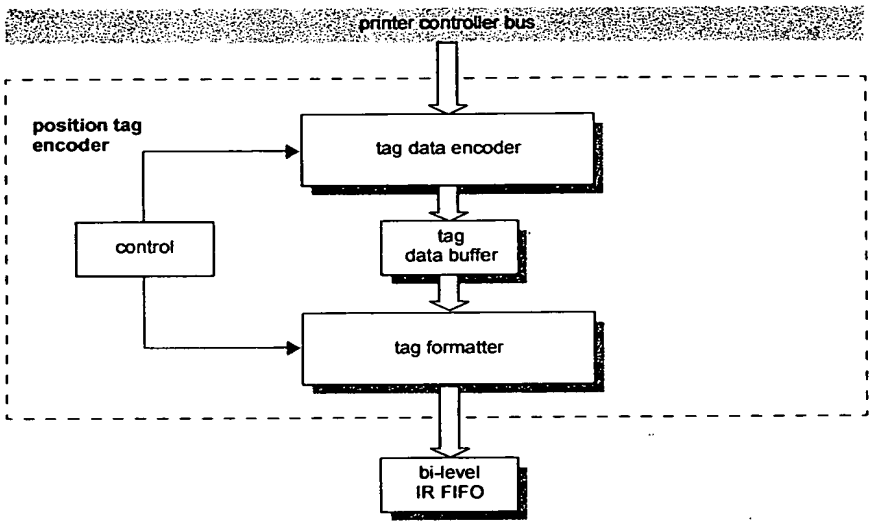


Figure 55. Two stages of position tag encoder

The position tag encoder provides full macrodot-to-dot conversion, thereby eliminating any special knowledge of macrodots from the bi-level FIFO user. However, the position tag encoder itself does make use of the redundancy caused by dot replication in macrodot

formation with respect to the dot axis only. Since the macrodot-to-dot ratio is 1:4, the position tag encoder operates at 1/4 of the dot frequency (60 MHz).

19.3.1 Tag Data Buffer

The tag data buffer consists of two identical buffers. One is read by the tag formatter while the other is being written to by the tag data encoder. The two buffers are then logically swapped.

Each buffer contains $256T$ bits, where T is the maximum number of tags in a given printed line of dots. The maximum amount of memory required for each buffer is 16KBytes (when $T = 512$). To support a 12-inch printhead, $T = 283$, so the memory required for each buffer is 9056 bytes (55% of the maximum). Note that the remainder of the position tag encoder does not change in size based on T.

As the encoded bits are generated by the tag data encoder, they are written to consecutive bits in the tag data buffer. The data bits are read out in random-access fashion by the tag formatter. Having 256 bits allocated to each tag allows the tag number to be used for generating the high bits of the address and the low 8 bits of the address to come from the sub-tag address generator.

The tag data buffer therefore has the structure as shown by Figure 56:

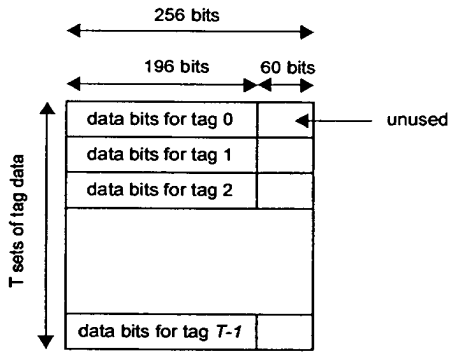


Figure 56. Tag data buffer logical structure

19.3.2 Tag Data Encoder

It is the responsibility of the tag data encoder to produce the 28 bits of encoded mode data and 168 bits of encoded tag data for a total of 196 bits of encoded data. The encoding scheme for position tags is defined in Section 16.

Each tag is represented on the page by a 17×17×4×4 structure of dots. However, since the tag data formatter is running at only 1/4 of the dot frequency, this results in a time of 1156 cycles (17×17×4) per tag.

The tag data encoder and control block functions are provided by a simple microprocessor core running at 60 MHz. This is desirable compared to the design effort to implement a specific Reed-Solomon encoder for two types of encoding and specific tag codeword construction.

The 196 bits of encoded tag data are written to one of the tag data buffers while the other is being read from by the tag formatter function.

The coordinates encoded within the tags for a given line of tags will depend on the width of the printhead and whether or not the pages are being printed in portrait or landscape mode, as shown in Figure 57.

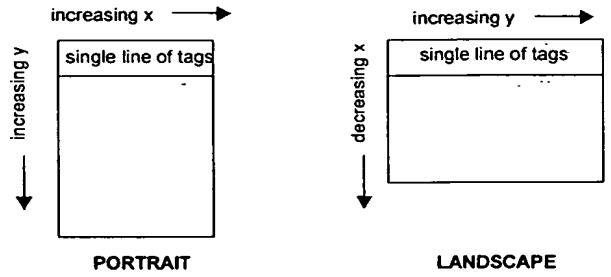


Figure 57. Relationship of coordinates to tags in a tag line

- If processing a *portrait* page, each printed line corresponds to changes in x coordinates. A printed line of tags therefore has a constant y coordinate, a start x coordinate of *0*, and end x coordinate of *T-1* with a Δx of *1*. To advance to the next line of tags, the y coordinate increases by 1 and x is reset to 0.
- If processing a *landscape* page, each printed line corresponds to changes in y coordinates. A printed line of tags therefore has a constant x coordinate, a start y coordinate of 0, and an end y coordinate of *T-1* with a Δy of *1*. To advance to the next line of tags, the x coordinate is decreased by 1 and y is reset to 0.

The high level control function for coordinate generation is outlined here in pseudocode.

```
if portraitMode
   For tagline = 0 to maxTagLines
      For tag = 0 to lastTagInLine
         EncodeTag(tag, pageId, tag, tagLine)
      EndFor
   EndFor
else
   tagX = maxTagLines
   For tagline = 0 to maxTagLines
      For tag = 0 to lastTagInLine
         EncodeTag(tag, pageId, tagX, tag)
         tagX = tagX - 1
      EndFor
   EndFor
EndIf
```

In relation to the EncodeTag function, an examination of the tag structure reveals:

- of the 168 data bits, 90 bits are copied to a maximum of 108 bits in the output with only trivial change. The remaining bits are generated check words based on GF(64).

- of the 28 mode bits, 8 bits are directly copied to the output. The remaining 20 are generated check words based on GF(16).

Therefore the 8 mode bits output and at least 90 bits of the data bits output can be output at a rate approximating 1 cycle per bit. The remaining 70 bits must be generated using Reed-Solomon encoding. This allows approximately 1000 cycles for 70 bits, or approximately 14 cycles per encoded bit. Encoding over GF(16) and GF(64) can be accomplished using small tables totalling 64 and 256 bytes respectively.

The bits are stored in the tag data buffer in bit generation order in order to simplify the writing process. The reading process (in the tag formatter) has random access to the generated bits, but has specific addressing hardware to assist in this task. The order for writing bits is as follows:

- 90-108 bits of data (15-18 × 6-bit codewords, depending on whether 1s or 0s had to be inserted to generate non 000000 and non 111111 codewords) based on the 90 bits of data (64 bits of pageId, 9 bits of tagX, 9 bits of tagY, 1 bit from the active area bitmap at tag resolution, 1 bit from the pressure sensitive area bitmap at tag resolution and 6 reserved bits).
- 78-60 bits of check words (13-10 × 6-bit check words, depending on how many data codewords were written)
- 8 bits of mode data (2 × 4-bit codewords)
- 20 bits of check words (5 × 4-bit check words)

19.3.3 Tag Formatter

The tag formatter is responsible for merging the encoded tag data with the tag structure, and placing the dots in the IR buffer in the correct order for printing. The encoded tag data is read from the tag data buffer as previously generated by the tag data encoder. The formatted dots are placed in the bi-level IR buffer such that the same data is clocked in 4 times. This allows the tag formatter to run at 1/4 of the dot rate.

We use a simple set of counters for formatting a set of 68 lines of dots (a set of 68 lines equates to 17 rows of 4-dot macrodots). The logic for the 68 lines of dots can be repeated until the page has finished printing. The within-tag counters index into an address table of 9-bit entries with the following attributes:

- If the entry's high bit = 0, then the macrodot comes from an encoded data bit. The encoded data bit to use is given by the concatenation of the current tag# (10 bits) and 8 bits of address from the low 8-bits of the entry.
- If the entry's high bit = 1, then the macrodot comes from the constant structure of a position tag (the bull's-eye and orientation bits). The entry's low bit determines the state of the macrodot.

The address table defines the structure of a positional tag at the macrodot level. It therefore has a constant size of 289 × 9-bit entries. The stored 8-bit addresses simply combine the tag encoding structure with the order that the bits are written to the tag data buffer. For the constant part of the tag (the bull's-eye and orientation bits), the entries are simply 100000000 for a white dot and 100000001 for a black dot.

The tag formatter is shown in pseudocode form:

```
For line = 0 to 16
```

```
   For dotY = 0 to 3
      For tag = 0 to numTagsInLine
         For pixel = 0 to 16
            adr = TableLookup[line, pixel]
            If (adr5=0)
               bit = TagDataBuffer[tag | adr0-4]
            Else
               bit = adr0
            EndIf
            Place 4 copies of bit in FIFO
         EndFor
      EndFor
   EndFor
EndFor
Swap TagDataBuffers
```

The pseudocode is readily transferred to logic, as illustrated by Figure 58.

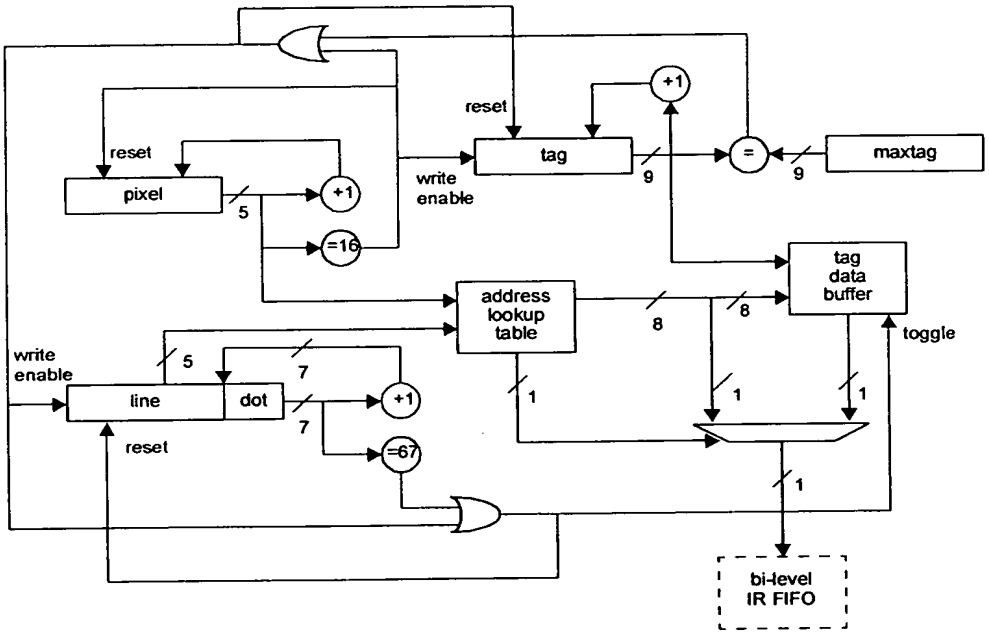


Figure 58. Tag formatter

20 Print Engine Driver

This section discusses rasterizing pages to the internal compressed page format expected by the print engine, as defined in Section 18, in terms of the compressed page driver which hides this device-dependent behavior from the higher-level raster image processor (RIP). The relationship between the RIP, the graphics system, the compressed page driver, and the print engine, is illustrated in Figure 59.

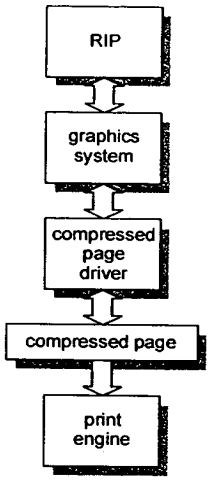


Figure 59. RIP and print engine driver

The RIP and the graphics system are considered generic components and are not defined further at this stage.

20.1 Graphics and Imaging Model

The compressed page driver is closely coupled with the graphics system used by the RIP, so that the driver can provide device-specific handling for different graphics and imaging operations, in particular compositing operations and text operations.

The graphics system renders images and graphics to a nominal resolution specified by the compressed page driver, but allows the compressed page driver to take control of rendering text. In particular, the graphics system provides sufficient information to the compressed page driver to allow it to render and position text at a higher resolution than the nominal device resolution.

The host graphics system requires random access to a contone page buffer at the nominal device resolution, into which it composites graphics and imaging objects, but it allows the compressed page driver to take control of the actual compositing - i.e. the compressed page driver manages the page buffer.

20.2 Two-Layer Page Buffer

The compressed page format contains a 267 ppi contone layer and an 800 dpi black layer. The black layer is conceptually *above* the contone layer, i.e. the black layer is composited *over* the contone layer by the print engine. The compressed page driver therefore main-

tains a page buffer which correspondingly contains a medium-resolution contone layer and a high-resolution black layer.

The graphics systems renders and composites objects into the page buffer bottom-up - i.e. later objects obscure earlier objects. This works naturally when there is only a single layer, but not when there are two layers which will be composited later. It is therefore necessary to detect when an object being placed on the contone layer obscures something on the black layer.

When obscuration is detected, the obscured black pixels are composited with the contone layer and removed from the black layer. The obscuring object is then laid down on the contone layer, possibly interacting with the black pixels in some way. If the compositing mode of the obscuring object is such that no interaction with the background is possible, then the black pixels can simply be discarded without being composited with the contone layer. In practice, of course, there is little interaction between the contone layer and the black layer, since images and text rarely overlap.

The compressed page driver specifies a nominal page resolution of 267 ppi to the graphics system. Where possible the compressed page driver relies on the graphics system to render image and graphics objects to the pixel level at 267 ppi, with the exception of *black* text. The compressed page driver fields all text rendering requests, detects and renders black text at 800 dpi, but returns non-black text rendering requests to the graphics system for rendering at 267 ppi.

Ideally the graphics system and the compressed page driver manipulate color in device-independent RGB, deferring conversion to device-specific CMY until the page is complete and ready to be sent to the printer. This reduces page buffer requirements and makes compositing more rational. Compositing in in a device-dependent color space is not ideal.

Ultimately the graphics system asks the compressed page driver to composite each rendered object into the compressed page driver's page buffer. Each such object uses 24-bit contone RGB, and has an explicit (or implicitly opaque) opacity channel.

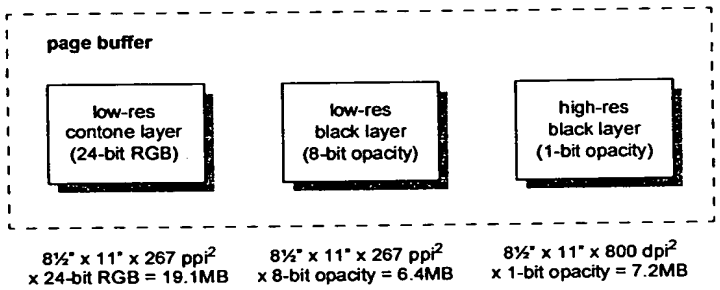


Figure 60. Two-layer page buffer

The compressed page driver maintains the two-layer page buffer in three parts. The first part is the medium-resolution (267 ppi) contone layer. This consists of a 24-bit RGB bitmap. The second part is a medium-resolution black layer. This consists of an 8-bit opacity bitmap. The third part is a high-resolution (800 dpi) black layer. This consists of a 1-bit opacity bitmap. The medium-resolution black layer is a subsampled version of the high-resolution opacity layer. In practice, assuming the medium resolution is an integer factor n of the high resolution (e.g. $n = 800 / 267 = 3$), each medium-resolution opacity

value is obtained by averaging the corresponding $n \times n$ high-resolution opacity values. This corresponds to box-filtered subsampling. The subsampling of the black pixels effectively antialiases edges in the high-resolution black layer, thereby reducing ringing artifacts when the contone layer is subsequently JPEG-compressed and decompressed.

The structure and size of the page buffer is illustrated in Figure 60.

20.3 Compositing Model

For the purposes of discussing the page buffer compositing model, we define the following variables.

Table 18. Compositing variables

variable	description	resolution	format
n	medium to high resolution scale factor	-	-
C_C	contone layer color	medium	8-bit color component
C_G	contone object color	medium	8-bit color component
α_G	contone object opacity	medium	8-bit opacity
α_L	medium-resolution black layer opacity	medium	8-bit opacity
α_B	black layer opacity	high	1-bit opacity
α_T	black object opacity	high	1-bit opacity

When a black object of opacity α_T is composited with the black layer, the black layer is updated as follows:

$$\alpha_{B_{x,y}} = \alpha_{B_{x,y}} \vee \alpha_{T_{x,y}} \quad (1)$$

$$\alpha_{L_{x,y}} = \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} 255\alpha_{B_{nx+i,ny+j}} \quad (2)$$

The object opacity is simply *or*ed with the black layer opacity (Eq. 1), and the corresponding part of the medium-resolution black layer is re-computed from the high-resolution black layer (Eq. 2).

When a contone object of color C_G and opacity α_G is composited with the contone layer, the contone layer and the black layer are updated as follows:

$$C_{C_{x,y}} = C_{C_{x,y}}(1 - \alpha_{L_{x,y}}) \text{ if } (\alpha_{G_{x,y}} > 0) \quad (3)$$

$$\alpha_{L_{x,y}} = 0 \text{ if } (\alpha_{G_{x,y}} > 0) \quad (4)$$

$$\alpha_{B_{x,y}} = 0 \text{ if } (\alpha_{G_{x/n,y/n}} > 0) \quad (5)$$

$$C_{C_{x,y}} = C_{C_{x,y}}(1 - \alpha_{G_{x,y}}) + C_{G_{x,y}}\alpha_{G_{x,y}} \quad (6)$$

Wherever the contone object hides the black layer, even if not fully opaquely, the affected black layer pixels are composited with the contone layer (Eq. 3) and removed from the

black layer (Eq. 4 and Eq. 5). The contone object is then composited with the contone layer (Eq. 6).

20.4 PAGE COMPRESSION

Once page rendering is complete, the compressed page driver converts the contone layer to printer-specific CMY with the help of color management functions provided by the graphics system.

The compressed page driver then compresses and packages the black layer and the contone layer into the compressed page format defined in Section 18. This is subsequently expanded in real time by the print engine.

The forward discrete cosine transform (DCT) is the costliest part of JPEG compression. In current high-quality software implementations, the forward DCT of each 8×8 block requires 12 integer multiplications and 32 integer additions [53]. This places a trivial additional load on each RIP DSP.

MEMJET PRINTHEAD

21 Memjet Printhead

A Memjet printhead is a drop-on-demand 1600 dpi inkjet printer that produces bi-level dots in up to 4 colors to produce a printed page of a particular width. Since the printhead prints dots at 1600 dpi, each dot is approximately 22.5μm in diameter, and spaced 15.875μm apart. Because the printing is bi-level, the input image should be dithered or error-diffused for best results.

Typically a Memjet printhead for a particular application is page-width. This enables the printhead to be stationary and allows the paper to move past the printhead. Figure 61 illustrates a typical configuration.

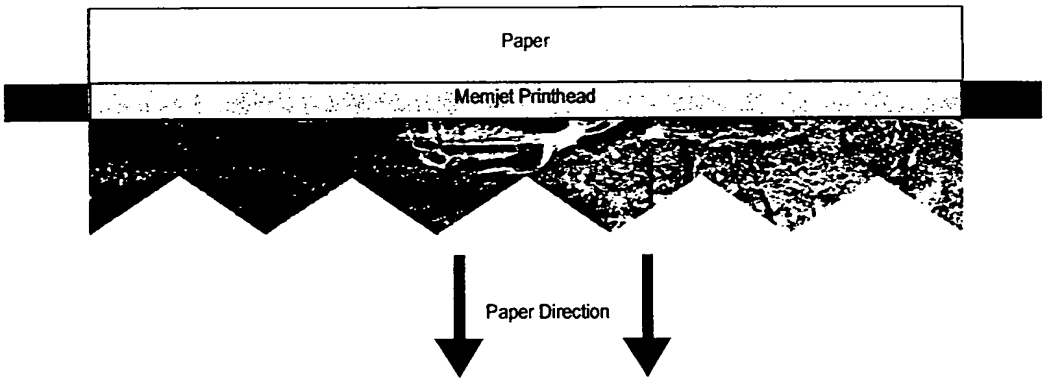


Figure 61. A Memjet printhead

A Memjet printhead is composed of a number of identical 1/2 inch Memjet segments. The segment is therefore the basic building block for constructing a printhead.

21.1 The Structure of a Memjet Segment

This section examines the structure of a single segment, the basic building block for constructing Memjet printheads.

21.1.1 Grouping of Nozzles Within a Segment

The nozzles within a single segment are grouped for reasons of physical stability as well as minimization of power consumption during printing. In terms of physical stability, a total of 10 nozzles share the same ink reservoir. In terms of power consumption, groupings are made to enable a low-speed and a high-speed printing mode.

Memjet segments support two printing speeds to allow speed/power consumption trade-offs to be made in different product configurations.

In the low-speed printing mode, 4 nozzles of each color are fired from the segment at a time. The exact number of nozzles fired depends on how many colors are present in the printhead. In a four color (e.g. CMYK) printing environment this equates to 16 nozzles firing simultaneously. In a three color (e.g. CMY) printing environment this equates to 12 nozzles firing simultaneously. To fire all the nozzles in a segment, 200 different sets of nozzles must be fired.

In the high-speed printing mode, 8 nozzles of each color are fired from the segment at a time. The exact number of nozzles fired depends on how many colors are present in the printhead. In a four color (e.g. CMYK) printing environment this equates to 32 nozzles firing simultaneously. In a three color (e.g. CMY) printing environment this equates to 24 nozzles firing simultaneously. To fire all the nozzles in a segment, 100 different sets of nozzles must be fired.

The power consumption in the low-speed mode is half that of the high-speed mode. Note, however, that the energy consumed to print a page is the same in both cases.

21.1.1.1 10 Nozzles Make a Pod

A single pod consists of 10 nozzles sharing a common ink reservoir. 5 nozzles are in one row, and 5 are in another. Each nozzle produces dots 22.5µm in diameter spaced on a 15.875µm grid to print at 1600 dpi. Figure 62 shows the arrangement of a single pod, with the nozzles numbered according to the order in which they must be fired.

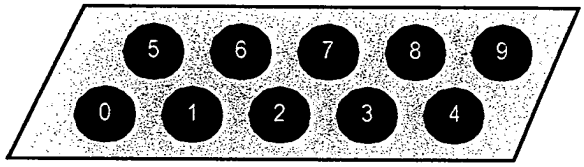


Figure 62. A single pod, numbered by firing order

Although the nozzles are fired in this order, the relationship of nozzles and physical placement of dots on the printed page is different. The nozzles from one row represent the even dots from one line on the page, and the nozzles on the other row represent the odd dots from the adjacent line on the page. Figure 63 shows the same pod with the nozzles numbered according to the order in which they must be loaded.

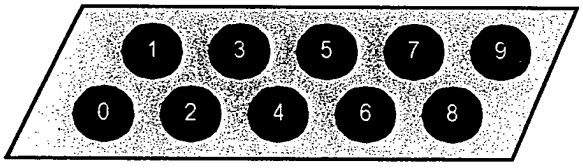


Figure 63. A single pod, numbered by load order

The nozzles within a pod are therefore logically separated by the width of 1 dot. The exact distance between the nozzles will depend on the properties of the Memjet firing mechanism. The printhead is designed with staggered nozzles designed to match the flow of paper.

21.1.1.2 1 Pod of Each Color Makes a Chromapod

One pod of each color are grouped together into a ***chromapod.*** The number of pods in a chromapod will depend on the particular application. In a monochrome printing system (such as one that prints only black), there is only a single color and hence a single pod. Photo printing application printheads require 3 colors (cyan, magenta, yellow), so Memjet segments used for these applications will have 3 pods per chromapod (one pod of each color). The expected maximum number of pods in a chromapod is 4, as used in a CMYK (cyan, magenta, yellow, black) printing system (such as a desktop printer). This maximum of 4 colors is not imposed by any physical constraints - it is merely an expected maximum

from the expected applications (of course, as the number of colors increases the cost of the segment increases and the number of these larger segments that can be produced from a single silicon wafer decreases).

A chromapod represents different color components of the same horizontal set of 10 dots on different lines. The exact distance between different color pods depends on the Memjet operating parameters, and may vary from one Memjet design to another. The distance is considered to be a constant number of dot-widths, and must therefore be taken into account when printing: the dots printed by the cyan nozzles will be for different lines than those printed by the magenta, yellow or black nozzles. The printing algorithm must allow for a variable distance up to about 8 dot-widths between colors. Figure 64 illustrates a single chromapod for a CMYK printing application.

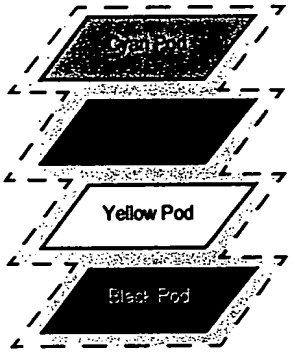


Figure 64. A Single Chromapod Contains 1 Pod of each Color

21.1.1.3 5 Chromapods Make a Podgroup

5 chromapods are organized into a single ***podgroup***. A podgroup therefore contains 50 nozzles for each color. The arrangement is shown in Figure 65, with chromapods numbered 0-4 and using a CMYK chromapod as the example. Note that the distance between adjacent chromapods is exaggerated for clarity.

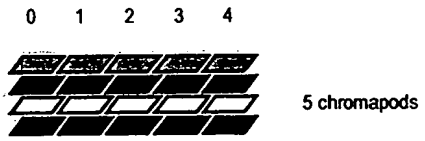


Figure 65. A Single Podgroup Contains 5 Chromapods

21.1.1.4 2 Podgroups Make a Phasegroup

2 podgroups are organized into a single ***phasegroup***. The phasegroup is so named because groups of nozzles within a phasegroup are fired simultaneously during a given firing phase (this is explained in more detail below). The formation of a phasegroup from 2 podgroups is entirely for the purposes of low-speed and high-speed printing via 2 PodgroupEnable lines.

During low-speed printing, only one of the two PodgroupEnable lines is set in a given firing pulse, so only one podgroup of the two fires nozzles. During high-speed printing, both PodgroupEnable lines are set, so both podgroups fire nozzles. Consequently a low-speed print takes twice as long as a high-speed print, since the high-speed print fires twice as many nozzles at once.

Figure 66 illustrates the composition of a phasegroup. The distance between adjacent podgroups is exaggerated for clarity.

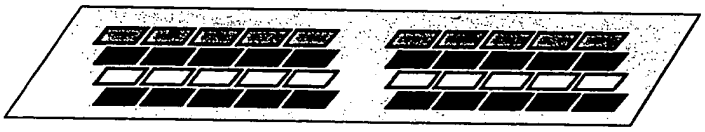


Figure 66. A single phasegroup contains 2 podgroups

21.1.1.5 2 Phasegroups Make a Firegroup

Two phasegroups (PhasegroupA and PhasegroupB) are organized into a single ***firegroup***, with 4 firegroups in each segment. Firegroups are so named because they all fire the same nozzles simultaneously. Two enable lines, AEnable and BEnable, allow the firing of PhasegroupA nozzles and PhasegroupB nozzles independently as different firing phases. The arrangement is shown in Figure 67. The distance between adjacent groupings is exaggerated for clarity.

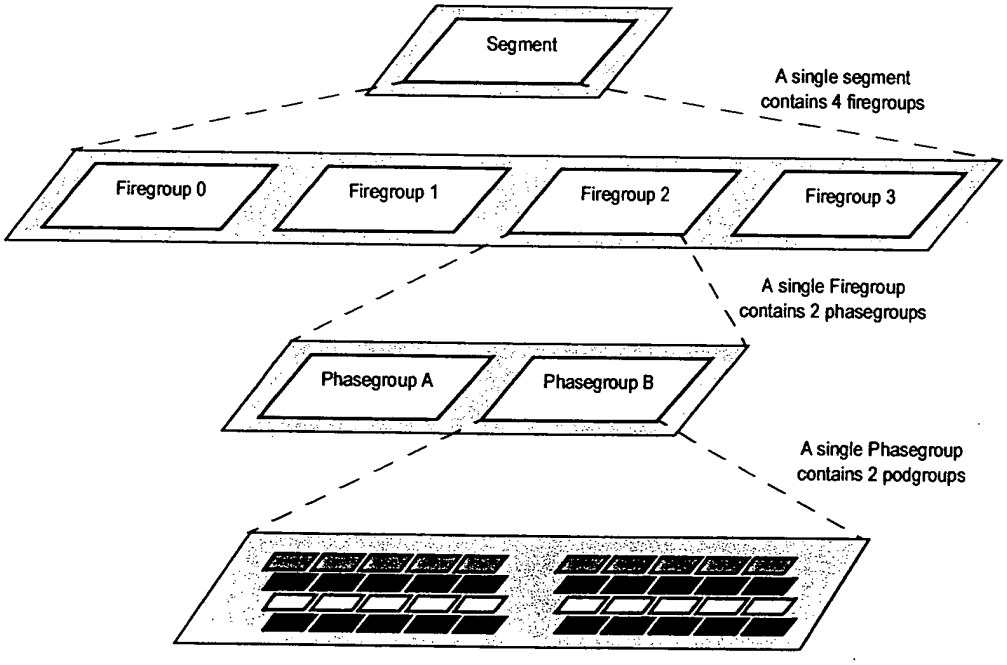


Figure 67. Relationship between Segments, Firegroups, Phasegroups, Podgroups and ChromaPods

21.1.1.6 Nozzle Grouping Summary

Table 19 is a summary of the nozzle groupings in a segment assuming a CMYK chromapod.

Table 19. Nozzle Groupings for a single segment

Name of Grouping	Composition	Replication Ratio	Nozzle Count
Nozzle	Base unit	1:1	1
Pod	Nozzles per pod	10:1	10
Chromapod	Pods per chromapod	C:1	10C
Podgroup	Chromapods per podgroup	5:1	50C
Phasegroup	Podgroups per phasegroup	2:1	100C
Firegroup	Phasegroups per firegroup	2:1	200C
Segment	Firegroups per segment	4:1	800C

The value of C, the number of colors contained in the segment, determines the total number of nozzles.

- With a 4 color segment, such as CMYK, the number of nozzles per segment is 3,200.
- With a 3 color segment, such as CMY, the number of nozzles per segment is 2,400.
- In a monochrome environment, the number of nozzles per segment is 800.

21.1.2 Load and Print Cycles

A single segment contains a total of 800C nozzles, where C is the number of colors in the segment. A *Print Cycle* involves the firing of up to all of these nozzles, dependent on the information to be printed. A *Load Cycle* involves the loading up of the segment with the information to be printed during the subsequent Print Cycle.

Each nozzle has an associated ***NozzleEnable*** bit that determines whether or not the nozzle will fire during the Print Cycle. The NozzleEnable bits (one per nozzle) are loaded via a set of shift registers.

Logically there are C shift registers per segment (one per color), each 800 deep. As bits are shifted into the shift register for a given color they are directed to the lower and upper nozzles on alternate pulses. Internally, each 800-deep shift register is comprised of two 400-deep shift registers: one for the upper nozzles, and one for the lower nozzles. Alternate bits are shifted into the alternate internal registers. As far as the external interface is concerned however, there is a single 800 deep shift register.

Once all the shift registers have been fully loaded (800 load pulses), all of the bits are transferred in parallel to the appropriate NozzleEnable bits. This equates to a single parallel transfer of 800C bits. Once the transfer has taken place, the Print Cycle can begin. The Print Cycle and the Load Cycle can occur simultaneously as long as the parallel load of all NozzleEnable bits occurs at the end of the Print Cycle.

21.1.2.1 Load Cycle

The Load Cycle is concerned with loading the segment's shift registers with the next Print Cycle's NozzleEnable bits.

Each segment has C inputs directly related to the C shift registers (where C is the number of colors in the segment). These inputs are named ***ColorNData***, where N is 1 to C (for example, a 4 color segment would have 4 inputs labeled ***Color1Data***, ***Color2Data***, ***Color3Data*** and ***Color4Data***). A single pulse on the ***SRClock*** line transfers C bits into the appropriate shift registers. Alternate pulses transfer bits to the lower and upper nozzles respectively. A total of 800 pulses are required for the complete transfer of data. Once all 800C bits have been transferred, a single pulse on the ***PTransfer*** line causes the parallel transfer of data from the shift registers to the appropriate NozzleEnable bits.

The parallel transfer via a pulse on PTransfer must take place *after* the Print Cycle has finished. Otherwise the NozzleEnable bits for the line being printed will be incorrect.

It is important to note that the odd and even dot outputs, although printed during the same Print Cycle, do not appear on the same physical output line. The physical separation of odd and even nozzles within the printhead, as well as separation between nozzles of different colors ensures that they will produce dots on different lines of the page. This relative difference must be accounted for when loading the data into the printhead. The actual difference in lines depends on the characteristics of the inkjet mechanism used in the printhead. The differences can be defined by variables D_1 and D_2 where D_1 is the distance between nozzles of different colors, and D_2 is the distance between nozzles of the same color. Table 20 shows the dots transferred to a C color segment on the first 4 pulses.

Table 20. Order of Dots Transferred to a Segment

Pulse	Dot	Color1 Line	Color2 Line	Color3 Line	ColorC Line
1	0	N	$N+D_1$[a]	$N+2D_1$	$N+(C-1)D_1$
2	1	$N+D_2$[b]	$N+D_1+D_2$	$N+2D_1+D_2$	$N+(C-1)D_1+D_2$
3	2	N	$N+D_1$	$N+2D_1$	$N+(C-1)D_1$
4	3	$N+D_2$	$N+D_1+D_2$	$N+2D_1+D_2$	$N+(C-1)D_1+D_2$

a. D_1 = number of lines between the nozzles of one color and the next (likely = 4-8)

b. D_2 = number of lines between two rows of nozzles of the same color (likely = 1)

And so on for all 800 pulses.

Data can be clocked into a segment at a maximum rate of 20 MHz, which will load the entire 800C bits of data in 40µs.

21.1.2.2 Print Cycle

A single Memjet printhead segment contains 800 nozzles. To fire them all at once would consume too much power and be problematic in terms of ink refill and nozzle interference. This problem is made more apparent when we consider that a Memjet printhead is composed of multiple 1/2 inch segments, each with 800 nozzles. Consequently two firing modes are defined: a low-speed printing mode and a high-speed printing mode:

- In the low-speed print mode, there are 200 phases, with each phase firing 4C nozzles (C per firegroup, where C is the number of colors).
- In the high-speed print mode, there are 100 phases, with each phase firing 8C nozzles, (2C per firegroup, where C is the number of colors).

The nozzles to be fired in a given firing pulse are determined by

- 3 bits ***ChromapodSelect*** (select 1 of 5 chromapods from a firegroup)
- 4 bits ***NozzleSelect*** (select 1 of 10 nozzles from a pod)
- 2 bits of ***PodgroupEnable*** lines (select 0, 1, or 2 podgroups to fire)

When one of the PodgroupEnable lines is set, only the specified Podgroup's 4 nozzles will fire as determined by ChromapodSelect and NozzleSelect. When both of the PodgroupEnable lines are set, both of the podgroups will fire their nozzles. For the low-speed mode, two fire pulses are required, with PodgroupEnable = 10 and 01 respectively. For the high-speed mode, only one fire pulse is required, with PodgroupEnable = 11.

The duration of the firing pulse is given by the ***AEnable*** and ***BEnable*** lines, which fire the PhasegroupA and PhasegroupB nozzles from all firegroups respectively. The typical duration of a firing pulse is 1.3 - 1.8 µs. The duration of a pulse depends on the viscosity of the ink (dependent on temperature and ink characteristics) and the amount of power available to the printhead. See Section 21.1.3 on page 139 for details on feedback from the printhead in order to compensate for temperature change.

The AEnable and BEnable are separate lines in order that the firing pulses can overlap. Thus the 200 phases of a low-speed Print Cycle consist of 100 A phases and 100 B phases, effectively giving 100 sets of Phase A and Phase B. Likewise, the 100 phases of a high-speed print cycle consist of 50 A phases and 50 B phases, effectively giving 50 phases of phase A and phase B.

Figure 68 shows the AEnable and BEnable lines during a typical Print Cycle. In a high-speed print there are 50 2µs cycles, while in a low-speed print there are 100 2µs cycles.

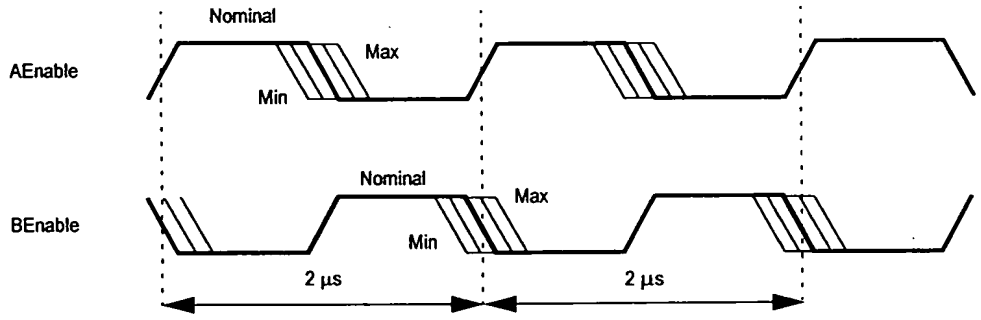


Figure 68. AEnable and BEnable During a Typical Print Cycle

For the high-speed printing mode, the firing order is:

- ChromapodSelect 0, NozzleSelect 0, PodgroupEnable 11 (Phases A and B)
- ChromapodSelect 1, NozzleSelect 0, PodgroupEnable 11 (Phases A and B)
- ChromapodSelect 2, NozzleSelect 0, PodgroupEnable 11 (Phases A and B)
- ChromapodSelect 3, NozzleSelect 0, PodgroupEnable 11 (Phases A and B)
- ChromapodSelect 4, NozzleSelect 0, PodgroupEnable 11 (Phases A and B)
- ChromapodSelect 0, NozzleSelect 1, PodgroupEnable 11 (Phases A and B)
- ...
- ChromapodSelect 3, NozzleSelect 9, PodgroupEnable 11 (Phases A and B)
- ChromapodSelect 4, NozzleSelect 9, PodgroupEnable 11 (Phases A and B)

For the low-speed printing mode, the firing order is similar. For each phase of the high speed mode where PodgroupEnable was 11, two phases of PodgroupEnable = 01 and 10 are substituted as follows:

- ChromapodSelect 0, NozzleSelect 0, PodgroupEnable 01 (Phases A and B)
- ChromapodSelect 0, NozzleSelect 0, PodgroupEnable 10 (Phases A and B)
- ChromapodSelect 1, NozzleSelect 0, PodgroupEnable 01 (Phases A and B)
- ChromapodSelect 1, NozzleSelect 0, PodgroupEnable 10 (Phases A and B)
- ...
- ChromapodSelect 3, NozzleSelect 9, PodgroupEnable 01 (Phases A and B)
- ChromapodSelect 3, NozzleSelect 9, PodgroupEnable 10 (Phases A and B)
- ChromapodSelect 4, NozzleSelect 9, PodgroupEnable 01 (Phases A and B)
- ChromapodSelect 4, NozzleSelect 9, PodgroupEnable 10 (Phases A and B)

When a nozzle fires, it takes approximately 100μs to refill. The nozzle cannot be fired before this refill time has elapsed. This limits the fastest printing speed to 100μs per line. In the high-speed print mode, the time to print a line is 100μs, so the time between firing a nozzle from one line to the next matches the refill time. The low-speed print mode is slower than this, so is also acceptable.

The firing of a nozzle also causes acoustic perturbations for a limited time within the common ink reservoir of that nozzle's pod. The perturbations can interfere with the firing of another nozzle within the same pod. Consequently, the firing of nozzles within a pod should be offset from each other as long as possible. We therefore fire four nozzles from a chromapod (one nozzle per color) and then move onto the next chromapod within the podgroup.

- In the low-speed printing mode the podgroups are fired separately. Thus the 5 chromapods within both podgroups must all fire before the first chromapod fires again, totalling 10 × 2μs cycles. Consequently each pod is fired once per 20μs.
- In the high-speed printing mode, the podgroups are fired together. Thus the 5 chromapods within a single podgroups must all fire before the first chromapod fires again, totalling 5 × 2μs cycles. Consequently each pod is fired once per 10 μs.

As the ink channel is 300μm long and the velocity of sound in the ink is around 1500m/s, the resonant frequency of the ink channel is 2.5MHz. Thus the low-speed mode allows 50 resonant cycles for the acoustic pulse to dampen, and the high-speed mode allows 25 resonant cycles. Consequently any acoustic interference is minimal in both cases.

21.1.3 Feedback from a Segment

A segment produces several lines of feedback. The feedback lines are used to adjust the timing of the firing pulses. Since multiple segments are collected together into a printhead, it is effective to share the feedback lines as a tri-state bus, with only one of the segments placing the feedback information on the feedback lines.

A pulse on the segment's ***SenseSegSelect*** line ANDed with data on ***Color1Data*** selects if the particular segment will provide the feedback. The feedback sense lines will come from that segment until the next SenseSegSelect pulse. The feedback sense lines are as follows:

- ***Tsense*** informs the controller how hot the printhead is. This allows the controller to adjust timing of firing pulses, since temperature affects the viscosity of the ink.
- ***Vsense*** informs the controller how much voltage is available to the actuator. This allows the controller to compensate for a flat battery or high voltage source by adjusting the pulse width.
- ***Rsense*** informs the controller of the resistivity (Ohms per square) of the actuator heater. This allows the controller to adjust the pulse widths to maintain a constant energy irrespective of the heater resistivity.
- ***Wsense*** informs the controller of the width of the critical part of the heater, which may vary up to ± 5% due to lithographic and etching variations. This allows the controller to adjust the pulse width appropriately.

21.1.4 Preheat Cycle

The printing process has a strong tendency to stay at the equilibrium temperature. To ensure that the first section of the printed photograph has a consistent dot size, the equilibrium temperature must be met *before* printing any dots. This is accomplished via a preheat cycle.

The Preheat cycle involves a single Load Cycle to all nozzles of a segment with 1s (i.e. setting all nozzles to fire), and a number of short firing pulses to each nozzle. The duration of the pulse must be insufficient to fire the drops, but enough to heat up the ink. Altogether about 200 pulses for each nozzle are required, cycling through in the same sequence as a standard Print Cycle.

Feedback during the Preheat mode is provided by Tsense, and continues until equilibrium temperature is reached (about 30° C above ambient). The duration of the Preheat mode is around 50 milliseconds, and depends on the ink composition.

Preheat is performed before each print job. This does not affect performance as it is done while the data is being transferred to the printer.

21.1.5 Cleaning Cycle

In order to reduce the chances of nozzles becoming clogged, a cleaning cycle can be undertaken before each print job. Each nozzle is fired a number of times into an absorbent sponge.

The cleaning cycle involves a single Load Cycle to all nozzles of a segment with 1s (i.e. setting all nozzles to fire), and a number of firing pulses to each nozzle. The nozzles are cleaned via the same nozzle firing sequence as a standard Print Cycle. The number of times that each nozzle is fired depends upon the ink composition and the time that the

printer has ben idle. As with preheat, the cleaning cycle has no effect on printer performance.

21.1.6 Printhead Interface Summary

Each segment has the following connections to the bond pads:

Table 21. Segment Interface Connections

Name	Lines	Description
Chromapod Select	3	Select which chromapod will fire (0-4)
NozzleSelect	4	Select which nozzle from the pod will fire (0-9)
PodgroupEnable	2	Enable the podgroups to fire (choice of: 01, 10, 11)
AEnable	1	Firing pulse for podgroup A
BEnable	1	Firing pulse for podgroup B
ColorNData	C	Input to shift registers (1 bit for each of C colors in the segment)
SRClock	1	A pulse on SRClock (ShiftRegisterClock) loads C bits from ColorData into the C shift registers.
PTransfer	1	Parallel transfer of data from the shift registers to the internal NozzleEnable bits (one per nozzle).
SenseSegSelect	1	A pulse on SenseSegSelect ANDed with data on Color1Data selects the sense lines for this segment.
Tsense	1	Temperature sense
Vsense	1	Voltage sense
Rsense	1	Resistivity sense
Wsense	1	Width sense
Logic GND	1	Logic ground
Logic PWR	1	Logic power
V-	21	Actuator Ground
V+	21	Actuator Power
TOTAL	62+C	(if C is 4, Total = 66)

21.2 MAKING MEMJET PRINTHEADS OUT OF SEGMENTS

A Memjet printhead is composed of a number of identical 1/2 inch printhead segments. These 1/2 inch segments are manufactured together or placed together after manufacture to produce a printhead of the desired length. Each 1/2 inch segments prints 800 1600 dpi bi-level dots in up to 4 colors over a different part of the page to produce the final image. Although each segment produces 800 dots of the final image, each dot is represented by a combination of colored inks.

A 4-inch printhead, for example, consists of 8 segments, typically manufactured as a monolithic printhead. In a typical 4-color printing application (cyan, magenta, yellow, black), each of the segments prints bi-level cyan, magenta, yellow and black dots over a different part of the page to produce the final image. The positions of the segments are shown in Figure 69.

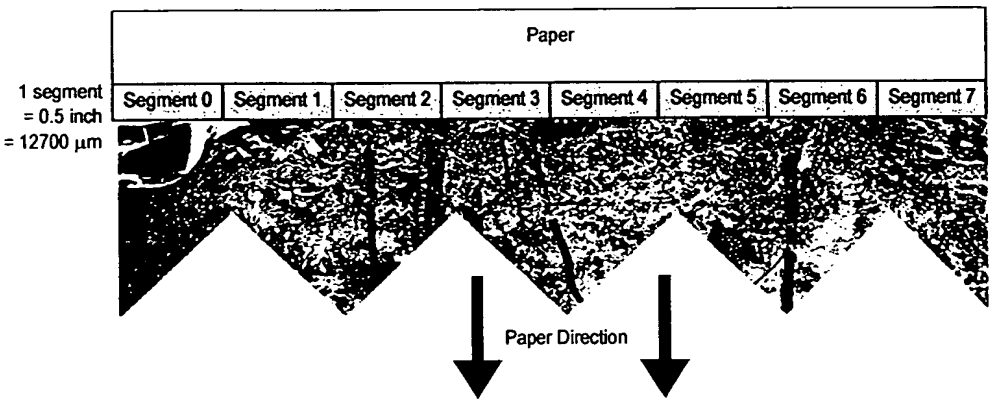


Figure 69. Arrangement of Segments in a 4-inch Printhead

An 8-inch printhead can be constructed from two 4-inch printheads or from a single 8-inch printhead consisting of 16 segments. Regardless of the construction mechanism, the effective printhead is still 8 inches in length.

A 2-inch printhead has a similar arrangement, but only uses 4 segments. Likewise, a full-bleed A4/Letter printer uses 17 segments for an effective 8½" printing area.

Since the total number of nozzles in a segment is 800C (see Table 19), the total number of nozzles in a given printhead with S segments is 800CS. Thus segment N is responsible for printing dots 800N to 800N+799.

A number of considerations must be made when wiring up a printhead. As the width of the printhead increases, the number of segments increases, and the number of connections also increases. Each segment has its own ColorData connections (C of them), as well as SRClock and other connections for loading and printing.

21.2.1 Loading Considerations

When the number of segments *S* is small it is reasonable to load all the segments simultaneously by using a common SRClock line and placing C bits of data on each of the ColorData inputs for the segments. In a 4-inch printer, S=8, and therefore the total number of bits to transfer to the printhead in a single SRClock pulse is 32. However for an 8-inch printer, S=16, and it is unlikely to be reasonable to have 64 data lines running from the print data generator to the printhead.

Instead, it is convenient to group a number of segments together for loading purposes. Each group of segments is small enough to be loaded simultaneously, and share an SRClock. For example, an 8-inch printhead can have 2 segment groups, each segment group containing 8 segments. 32 ColorData lines can be shared for both groups, with 2 SRClock lines, one per segment group.

When the number of segment groups is not easily divisible, it is still convenient to group the segments. One example is a 8½" printer for producing A4/Letter pages. There are 17 segments, and these can be grouped as two groups of 9 (9C bits of data going to each segment, with all 9C bits used in the first group, and only 8C bits used for the second group), or as 3 groups of 6 (again, C bits are unused in the last group).

As the number of segment groups increases, the time taken to load the printhead increases. When there is only one group, 800 load pulses are required (each pulse transfers C data bits). When there are G groups, 800G load pulses are required. The bandwidth of the connection between the data generator and the printhead must be able to cope and be within the allowable timing parameters for the particular application.

If *G* is the number of segment groups, and *L* is the largest number of segments in a group, the printhead requires LC ColorData lines and G SRClock lines. Regardless of G, only a single PTransfer line is required - it can be shared across all segments.

Since L segments in each segment group are loaded with a single SRClock pulse, any printing process must produce the data in the correct sequence for the printhead. As an example, when G=2 and L=4, the first SRClock1 pulse will transfer the ColorData bits for the next Print Cycle's dot 0, 800, 1600, and 2400. The first SRClock2 pulse will transfer the ColorData bits for the next Print Cycle's dot 3200, 4000, 4800, and 5600. The second SRClock1 pulse will transfer the ColorData bits for the next Print Cycle's dot 1, 801, 1601, and 2401. The second SRClock2 pulse will transfer the ColorData bits for the next Print Cycle's dot 3201, 4001, 4801 and 5601.

After 800G SRClock pulses (800 to each of SRClock1 and SRClock2), the entire line has been loaded into the printhead, and the common PTransfer pulse can be given.

It is important to note that the odd and even color outputs, although printed during the same Print Cycle, do not appear on the same physical output line. The physical separation of odd and even nozzles within the printhead, as well as separation between nozzles of different colors ensures that they will produce dots on different lines of the page. This relative difference must be accounted for when loading the data into the printhead. The actual difference in lines depends on the characteristics of the inkjet mechanism used in the printhead. The differences can be defined by variables D_1 and D_2 where D_1 is the distance between nozzles of different colors, and D_2 is the distance between nozzles of the same

color. Considering only a single segment group, Table 22 shows the dots transferred to segment *n* of a printhead during the first 4 pulses of the shared SRClock.

Table 22. Order of Dots Transferred to a Segment in a Printhead

Pulse	Dot	Color1 Line	Color2 Line	ColorC Line
1	800S[a]	N	N+D_1[b]	N+(C-1)D_1
2	800S+1	N+D_2[c]	N+D_1+D_2	N+(C-1)D_1+D_2
3	800S+2	N	N+D_1	N+(C-1)D_1
4	800S+3	N+D_2	N+D_1+D_2	N+(C-1)D_1+D_2

a. S = segment number

b. D_1 = number of lines between the nozzles of one color and the next (likely = 4-8)

c. D_2 = number of lines between two rows of nozzles of the same color (likely = 1)

And so on for all 800 SRClock pulses to the particular segment group.

21.2.2 Printing Considerations

With regards to printing, we print 4C nozzles from each segment in the low-speed printing mode, and 8C nozzles from each segment in the high speed printing mode.

While it is certainly possible to wire up segments in any way, this document only considers the situation where all segments fire simultaneously. This is because the low-speed printing mode allows low-power printing for small printheads (e.g. 2-inch and 4-inch), and the controller chip design assumes there is sufficient power available for the large print sizes (such as 8-18 inches). It is a simple matter to alter the connections in the printhead to allow grouping of firing should a particular application require it.

When all segments are fired at the same time 4CS nozzles are fired in the low-speed printing mode and 8CS nozzles are fired in the high-speed printing mode. Since all segments print simultaneously, the printing logic is the same as defined in Section 21.1.2.2 on page 136.

The timing for the two printing modes is therefore:

- 200 µs to print a line at low speed (comprised of 100 2µs cycles)
- 100 µs to print a line at high speed (comprised of 50 2µs cycles)

21.2.3 Feedback Considerations

A segment produces several lines of feedback, as defined in Section 21.1.3 on page 139. The feedback lines are used to adjust the timing of the firing pulses. Since multiple segments are collected together into a printhead, it is effective to share the feedback lines as a tri-state bus, with only one of the segments placing the feedback information on the feedback lines at a time.

Since the selection of which segment will place the feedback information on the shared Tsense, Vsense, Rsense, and Wsense lines uses the Color1Data line, the groupings of segments for loading data can be used for selecting the segment for feedback.

Just as there are G SRClock lines (a single line is shared between segments of the same segment group), there are G SenseSegSelect lines shared in the same way. When the correct SenseSegSelect line is pulsed, the segment of that group whose Color1Data bit is set will start to place data on the shared feedback lines. The segment previously active in terms of feedback must also be disabled by having a 0 on its Color1Data bit, and this segment may be in a different segment group. Therefore when there is more than one segment group. changing the feedback segment requires two steps: disabling the old segment, and enabling the new segment.

21.2.4 Printhead Connection Summary

This section assumes that a printhead has been constructed from a number of segments as described in the previous sections. It assumes that for data loading purposes, the segments have been grouped into G segment groups, with L segments in the largest segment group. It assumes there are C colors in the printhead. It assumes that the firing mechanism for the printhead is that all segments fire simultaneously, and only one segment at a time places feedback information on a common tri-state bus. Assuming all these things, Table 23 lists the external connections that are available from a printhead:

Table 23. Printhead Connections

Name	#Pins	Description
ChromapodSelect	3	Select which chromapod will fire (0-4)
NozzleSelect	4	Select which nozzle from the pod will fire (0-9)
PodgroupEnable	2	Enable the podgroups to fire (choice of: 01, 10, 11)
AEnable	1	Firing pulse for phasegroup A
BEnable	1	Firing pulse for phasegroup B
ColorData	CL	Inputs to C shift registers of segments 0 to L-1
SRClock	G	A pulse on SRClock[N] (ShiftRegisterClock N) loads the current values from ColorData lines into the L segments in segment group N.
PTransfer	1	Parallel transfer of data from the shift registers to the internal NozzleEnable bits (one per nozzle).
SenseSegSelect	G	A pulse on SenseSegSelect N ANDed with data on Color1Data[n] selects the sense lines for segment n in segment group N.
Tsense	1	Temperature sense
Vsense	1	Voltage sense
Rsense	1	Resistivity sense
Wsense	1	Width sense
Logic GND	1	Logic ground
Logic PWR	1	Logic power
V-	Bus bars	Actuator Ground
V+		Actuator Power
TOTAL	18+2G+CL	

22 Memjet Printhead Interface

The printhead interface (PHI) is the means by which the processor loads the Memjet printhead with the dots to be printed, and controls the actual dot printing process. The PHI contains:

- a LineSyncGen unit (LSGU), which provides synchronization signals for multiple chips (allows side-by-side printing and front/back printing) as well as stepper motors.
- a Memjet interface (MJI), which transfers data to the Memjet printhead, and controls the nozzle firing sequences during a print.
- a line loader/format unit (LLFU) which loads the dots for a given print line into local buffer storage and formats them into the order required for the Memjet printhead.

The units within the PHI are controlled by a number of registers that are programmed by the processor. In addition, the processor is responsible for setting up the appropriate parameters in the DMA controller for the transfers from memory to the LLFU. This includes loading white (all 0's) into appropriate colors during the start and end of a page so that the page has clean edges.

The PHI is capable of dealing with a variety of printhead lengths and formats. In terms of broad operating customizations, the PHI is parameterized as follows:

Table 24. Basic Printing Parameters

Name	Description	Range
MaxColors	No of Colors in printhead	1-4
SegmentsPerXfer	No of segments written to per transfer. Is equal to the number of segments in the largest segment group	1-9
SegmentGroups	No of segment groups in printhead	1-4

The internal structure of the PHI allows for a maximum of 4 colors, 9 segments per transfer, and 4 transfers. Transferring 4 colors to 9 segments is 36 bits per transfer, and 4 transfers to 9 segments equates to a maximum printed line length of 18 inches. The total number of dots per line printed by an 18-inch 4 color printhead is 115,200 (18 × 1600 × 4).

Other example settings are shown in Table 25:

Table 25. Example Settings for Basic Printing Parameters

Printer Length	Printer Type	MaxColors	SegmentsPerXfer	SegmentGroups	Bits per Xfer	Comments
4 inch CMY	Photo	3	8	1	24	
8 inch CMYK	A4/Letter	4	8	2	32	
8½ inch CMYK	A4/Letter full bleed	4	9	2	36	Last xfer not fully used
12 inch CMYK	A4 long / A3 short	4	8	3	32	
16 inch CMYK		4	8	4	32	
17 inch CMYK	A3 long full bleed	4	9	4	36	Last xfer not fully used
18 inch CMYK		4	9	4	36	

22.1 BLOCK DIAGRAM OF PRINTHEAD INTERFACE

The internal structure of the Printhead Interface is shown in Figure 70.

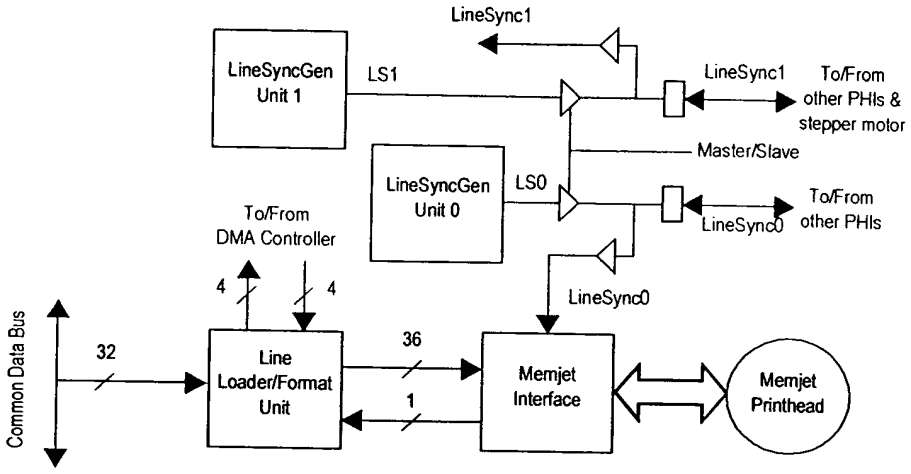


Figure 70. Internal Structure of Printhead Interface

In the PHI there are two LSGUs. The first LSGU produces LineSync0, which is used to control the Memjet Interface in all synchronized chips. The second LSGU produces LineSync1 which is used to pulse the paper drive stepper motor.

The Master/Slave pin on the chip allows multiple chips to be connected together for side-by-side printing, front/back printing etc. via a Master/Slave relationship. When the Master/Slave pin is attached to V_{DD}, the chip is considered to be the Master, and LineSync pulses generated by the two LineSyncGen units are enabled onto the two tri-state LineSync common lines (LineSync0 and LineSync1, shared by all the chips). When the Master/Slave pin is attached to GND, the chip is considered to be the Slave, and LineSync pulses generated by the two LineSyncGen units are not enabled onto the common LineSync lines. In this way, the Master chip's LineSync pulses are used by all PHIs on all the connected chips.

The following sections detail the LineSyncGen Unit, the Line Loader/Format Unit and Memjet Interface respectively.

22.2 LINESYNCGEN UNIT

The LineSyncGen units (LSGU) are responsible for generating the synchronization pulses required for printing a page. Each LSGU produces an external LineSync signal to enable line synchronization. The generator inside the LGSU generates a LineSync pulse when told to 'go', and then every so many cycles until told to stop. The LineSync pulse defines the start of the next line.

The exact number of cycles between LineSync pulses is determined by the CyclesBetweenPulses register, one per generator. It must be at least long enough to allow one line to print (100 µs or 200 µs depending on whether the speed is low or high) and another line to load, but can be longer as desired (for example, to accommodate special requirements of paper transport circuitry). If the CyclesBetweenPulses register is set to a number less than a line

print time, the page will not print properly since each LineSync pulse will arrive before the particular line has finished printing.

The following interface registers are contained in the LSGU:

Table 26. LineSyncGen Unit Registers

Register Name	Description
CyclesBetweenPulses	The number of cycles to wait between generating one Line-Sync pulse and the next.
Go	Controls whether the LSGU is currently generating LineSync pulses or not. A write of 1 to this register generates a LineSync pulse, transfers CyclesBetweenPulses to CyclesRemaining, and starts the countdown. When CyclesRemaining hits 0, another LineSync pulse is generated, CyclesBetweenPulses is transferred to CyclesRemaining and the countdown is started again. A write of 0 to this register stops the countdown and no more LineSync pulses are generated.
CyclesRemaining	A status register containing the number of cycles remaining until the next LineSync pulse is generated.

The LineSync pulse is not used directly from the LGSU. The LineSync pulse is enabled onto a tri-state LineSync line only if the Master/Slave pin is set to Master. Consequently the LineSync pulse is only used in the form as generated by the Master chip (pulses generated by Slave chips are ignored).

22.3 MEMJET INTERFACE

The Memjet interface (MJI) transfers data to the Memjet printhead, and controls the nozzle firing sequences during a print.

The MJI is simply a State Machine (see Figure 71) which follows the printhead loading and firing order described in Section 21.2.1 on page 142, Section 21.2.2 on page 143, and includes the functionality of the Preheat Cycle and Cleaning Cycle as described in Section 21.1.4 on page 139 and Section 21.1.5 on page 139. Both high-speed and low-speed printing modes are available, although the MJI always fires a given nozzle from all segments in a printhead simultaneously (there is no separate firing of nozzles from one segment and then others). Dot counts for each color are also kept by the MJI.

The MJI loads data into the printhead from a choice of 2 data sources:

- All 1s. This means that all nozzles will fire during a subsequent Print cycle, and is the standard mechanism for loading the printhead for a preheat or cleaning cycle.
- From the 36-bit input held in the Transfer register of the LLFU. This is the standard means of printing an image. The 36-bit value from the LLFU is directly sent to the printhead and a 1-bit 'Advance' control pulse is sent to the LLFU.

The MJI knows how many lines it has to print for the page. When the MJI is told to 'go', it waits for a LineSync pulse before it starts the first line. Once it has finished loading/printing a line, it waits until the next LineSync pulse before starting the next line. The MJI stops once the specified number of lines has been loaded/printed, and ignores any further LineSync pulses.

The MJI is therefore directly connected to the LLFU, LineSync0 (shared between all synchronized chips), and the external Memjet printhead. The basic structure is shown in Figure 71.

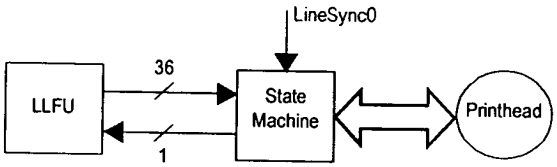


Figure 71. Memjet Interface

The MJI accepts 36 bits of data from the LLFU. Of these 36 bits, only the bits corresponding to the number of segments and number of colors will be valid. For example, if there are only 2 colors and 9 segments, bits 0-1 will be valid for segment 0, bits 2-3 will be invalid, bits 4-5 will be valid for segment 1, bits 6-7 will be invalid etc. The state machine does not care which bits are valid and which bits are not valid - it merely passes the bits out to the printhead. The data lines and control signals coming out of the MJI can be wired appropriately to the pinouts of the chip, using as few pins as required by the application range of the chip (see Section 22.3.1 on page 148 for more information).

22.3.1 Connections to Printhead

The MJI has a number of connections to the printhead, including a maximum of 4 colors, clocked in to a maximum of 9 segments per transfer to a maximum of 4 segment groups. The lines coming from the MJI can be directly connected to pins on the chip, although not all lines will always be pins. For example, if the chip is specifically designed for only connecting to 8 inch CMYK printers, only 32 bits of data need to be transferred each transfer pulse. Consequently 32 pins of data out (8 pins per color), and not 36 pins are required. In the same way, only 2 SRClock pulses are required, so only 2 pins instead of 4 pins are required to cater for the different SRClocks. And so on.

If the chip must be completely generic, then all connections from the MJI must be connected to pins on the chip (and thence to the Memjet printhead).

Table 27 lists the maximum connections from the MJI, many of which are always connected to pins on the chip. Where the number of pins is variable, a footnote explains what the number of pins depends upon. The sense of input and output is with respect to the MJI. The names correspond to the pin connections on the printhead.

Table 27. Memjet Interface Connections

Name	#Pins	I/O	Description
Chromapod Select	3	O	Select which chromapod will fire (0-4)
NozzleSelect	4	O	Select which nozzle from the pod will fire (0-9)
PodgroupEnable	2	O	Enable the podgroups to fire (choice of: 01, 10, 11)
AEnable	1	O	Firing pulse for podgroup A. In the current design all segments fire simultaneously, although multiple AEnable lines could be added for dividing the firing sequence over multiple segment groups for reasons of power and speed.

Table 27. Memjet Interface Connections

Name	#Pins	I/O	Description
BEnable	1	O	Firing pulse for podgroup B. In the current design all segments fire simultaneously, although multiple BEnable lines could be added for dividing the firing sequence over multiple segment groups for reasons of power and speed.
Color1Data[0-8]	9[a]	O	Output to Color1Data shift register of segments 0-8
Color2Data[0-8]	9[b]	O	Output to Color2Data shift register of segments 0-8
Color3Data[0-8]	9[c]	O	Output to Color3Data shift register of segments 0-8
Color4Data[0-8]	9[d]	O	Output to Color4Data shift register of segments 0-8
SRClock[1-4]	4[e]	O	A pulse on SRClock[N] (ShiftRegisterClock) loads the current values from Color1Data[0-8], Color2Data[0-8], Color3Data[0-8] and Color4Data[0-8] into the segment group N on the printhead.
PTransfer	1	O	Parallel transfer of data from the shift registers to the printhead's internal NozzleEnable bits (one per nozzle).
SenseSegSelect[1-4]	4[f]	O	A pulse on SenseSegSelect[N] ANDed with data on Color1Data[n] enables the sense lines for segment n in segment group N of the printhead.
Tsense	1	I	Temperature sense
Vsense	1	I	Voltage sense
Rsense	1	I	Resistivity sense
Wsense	1	I	Width sense
TOTAL	52		

a. Although 9 lines are available from the MJI, the number of pins coming from the chip will only reflect the actual number of segments in a segment group. The pins for Color1Data are mandatory, since each printhead must print in at *least* 1 color.

b. These lines are only translated into pins if the chip is to control a printhead with at least 2 colors. Although 9 lines are available from the MJI, the number of pins coming from the chip for Color2Data will only reflect the *actual* number of segments in a segment group.

c. These lines are only translated into pins if the chip is to control a printhead with at least 3 colors. Although 9 lines are available from the MJI, the number of pins coming from the chip for Color3Data will only reflect the *actual* number of segments in a segment group.

d. These lines are only translated into pins if the chip is to control a printhead with 4 colors. Although 9 lines are available from the MJI, the number of pins coming from the chip for Color4Data will only reflect the *actual* number of segments in a segment group.

e. Although 4 lines are available from the MJI, the number of pins coming from the chip will only reflect the *actual* number of segment groups. A minimum of 1 pin is required since there is at least 1 segment group (the entire printhead).

f. Although 4 lines are available from the MJI, the number of pins coming from the chip will only reflect the *actual* number of segment groups. A minimum of 1 pin is required since there is at least 1 segment group (the entire printhead).

22.3.2 Firing Pulse Duration

The duration of firing pulses on the AEnable and BEnable lines depend on the viscosity of the ink (which is dependant on temperature and ink characteristics) and the amount of power available to the printhead. The typical pulse duration range is 1.3 to 1.8 µs. The MJI therefore contains a programmable pulse duration table, indexed by feedback from the printhead. The table of pulse durations allows the use of a lower cost power supply, and aids in maintaining more accurate drop ejection.

The Pulse Duration table has 256 entries, and is indexed by the current Vsense and Tsense settings. The upper 4-bits of address come from Vsense, and the lower 4-bits of address

come from Tsense. Each entry is 8 bits, and represents a fixed point value in the range of 0-4μs. The process of generating the AEnable and BEnable lines is shown in Figure 72.

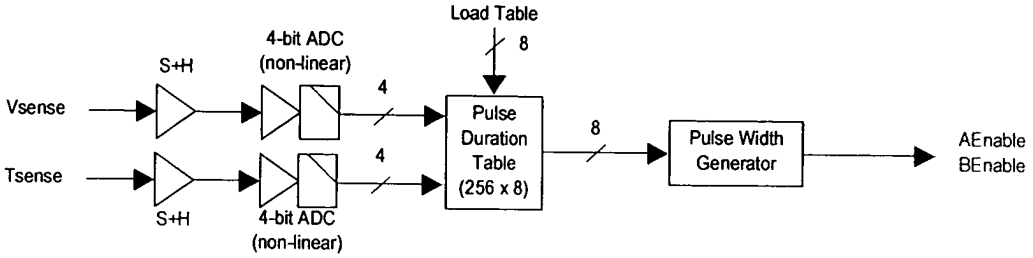


Figure 72. Generation of AEnable and BEnable Pulse Widths

The 256-byte table is written by the CPU before printing the first page. The table may be updated in between pages if desired. Each 8-bit pulse duration entry in the table combines:

- User brightness settings (from the page description)
- Viscosity curve of ink (from the QA Chip)
- Rsense
- Wsense
- Vsense
- Tsense

22.3.3 Dot Counts

The MJI maintains a count of the number of dots of each color fired from the printhead. The dot count for each color is a 32-bit value, individually cleared under processor control. At 32-bits length, each dot count can hold a maximum coverage dot count of 17 8-inch × 12-inch pages, although in typical usage, the dot count will be read and cleared after each page or half-page.

The dot counts are used by the processor to update the QA chip in order to predict when the ink cartridge runs out of ink. The processor knows the volume of ink in the cartridge for each of the colors from the QA chip. Counting the number of drops eliminates the need for ink sensors, and prevents the ink channels from running dry. An updated drop count is written to the QA chip after each page. A new page will not be printed unless there is enough ink left, and allows the user to change the ink without getting a dud half-printed page which must be reprinted.

The layout of the dot counter for Color1 is shown in Figure 73. The remaining 3 dot counters (Color1DotCount, Color2DotCount, and Color3DotCount) are identical in structure.

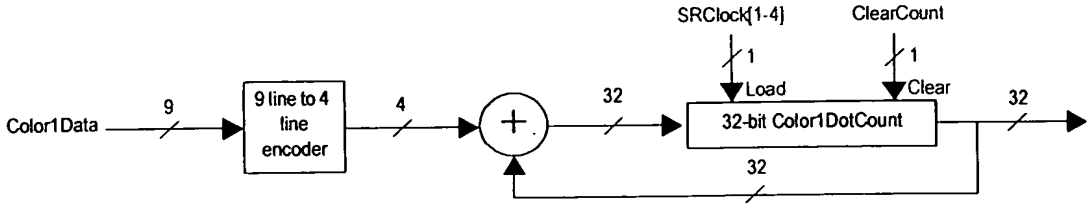


Figure 73. Dot Count Logic

22.3.4 Registers

The processor communicates with the MJI via a register set. The registers allow the processor to parameterize a print as well as receive feedback about print progress.

The following registers are contained in the MJI:

Table 28. Memjet Interface Registers

Register Name	Description
Print Parameters	
SegmentsPerXfer	The number of segments to write to each transfer. This also equals the number of cycles to wait between each transfer (before generating the next Advance pulse). Each transfer has MaxColors × SegmentsPerXfer valid bits.
SegmentGroups	The number of segment groups in the printhead. This equals the number of times that SegmentsPerXfer cycles must elapse before a single dot has been written to each segment of the printhead. The MJI does this 800 times to completely transfer all the data for the line to the printhead.
PrintSpeed	Whether to print at low or high speed (determines the value on the PodgroupEnable lines during the print).
NumLines	The number of Load/Print cycles to perform.
Monitoring the Print (read only from point of view of processor)	
Status	The Memjet Interface's Status Register
LinesRemaining	The number of lines remaining to be printed. Only valid while Go=1. Starting value is NumLines and counts down to 0.
TransfersRemaining	The number of sets of SegmentGroups transfers remaining before the Printhead is considered loaded for the current line. Starts at 800 and counts down to 0. Only valid while Go=1.
SegGroupsRemaining	The number of segment groups remaining in the current set of transfers of 1 dot to each segment. Starts at SegmentGroups and counts down to 0. Only valid while Go=1.
SenseSegment	The 9-bit value to place on the Color1Data lines during a subsequent feedback SenseSegSelect pulse. Only 1 of the 9 bits should be set, corresponding to one of the (maximum) 9 segments. See SenseSelect for how to determine which of the segment groups to sense.

Table 28. Memjet Interface Registers

Register Name	Description
SetAllNozzles	If non-zero, the 36-bit value written to the printhead during the LoadDots process is all 1s, so that all nozzles will be fired during the subsequent PrintDots process. This is used during the preheat and cleaning cycles. If 0, the 36-bit value written to the printhead comes from the LLFU. This is the case during the actual printing of regular images.
Actions	
Reset	A write to this register resets the MJI, stops any loading or printing processes, and loads all registers with 0.
SenseSelect	A write to this register with any value clears the FeedbackValid bit of the Status register, and the remaining action depends on the values in the LoadingDots and PrintingDots status bits. If either of the status bits are set, the Feedback bit is cleared and nothing more is done. If both status bits are clear, a pulse is given simultaneously on all 4 SenseSegSelect lines with all ColorData bits 0. This stops any existing feedback. Depending on the two low-order bits written to SenseSelect register, a pulse is given on SenseSegSelect1, SenseSegSelect2, SenseSegSelect3, or SenseSegSelect4 line, with the Color1Data bits set according to the SenseSegment register. Once the various sense lines have been tested, the values are placed in the Tsense, Vsense, Rsense, and Wsense registers, and the Feedback bit of the Status register is set.
Go	A write of 1 to this bit starts the LoadDots / PrintDots cycles, which commences with a wait for the first LineSync pulse. A total of NumLines lines are printed, each line being loaded/printed after the receipt of a LineSync pulse. The loading of each line consists of SegmentGroups 36-bit transfers. As each line is printed, LinesRemaining decrements, and TransfersRemaining is reloaded with SegmentGroups again. The status register contains print status information. Upon completion of NumLines, the loading/printing process stops, the Go bit is cleared, and any further LineSync pulses are ignored. During the final print cycle, nothing is loaded into the printhead. A write of 0 to this bit stops the print process, but does not clear any other registers.
ClearCounts	A write to this register clears the Color1DotCount, Color2DotCount, Color3DotCount, and Color4DotCount registers if bits 0, 1, 2, or 3 respectively are set. Consequently a write of 0 has no effect.
Feedback	
Tsense	Read only feedback of Tsense from the last SenseSegSelect pulse sent to segment SenseSegment. Is only valid if the FeedbackValid bit of the Status register is set.
Vsense	Read only feedback of Vsense from the last SenseSegSelect pulse sent to segment SenseSegment. Is only valid if the FeedbackValid bit of the Status register is set.
Rsense	Read only feedback of Rsense from the last SenseSegSelect pulse sent to segment SenseSegment. Is only valid if the FeedbackValid bit of the Status register is set.
Wsense	Read only feedback of Wsense from the last SenseSegSelect pulse sent to segment SenseSegment. Is only valid if the FeedbackValid bit of the Status register is set.
Color1DotCount	Read only 32-bit count of color1 dots sent to the printhead.
Color2DotCount	Read only 32-bit count of color2 dots sent to the printhead.

Table 28. Memjet Interface Registers

Register Name	Description
Color3DotCount	Read only 32-bit count of color3 dots sent to the printhead
Color4DotCount	Read only 32-bit count of color4 dots sent to the printhead

The MJI's Status Register is a 16-bit register with bit interpretations as follows:

Table 29. MJI Status Register

Name	Bits	Description
LoadingDots	1	If set, the MJI is currently loading dots, with the number of dots remaining to be transferred in TransfersRemaining. If clear, the MJI is not currently loading dots
PrintingDots	1	If set, the MJI is currently printing dots. If clear, the MJI is not currently printing dots.
PrintingA	1	This bit is set while there is a pulse on the AEnable line
PrintingB	1	This bit is set while there is a pulse on the BEnable line
FeedbackValid	1	This bit is set while the feedback values Tsense, Vsense, Rsense, and Wsense are valid.
Reserved	3	-
PrintingChromapod	4	This holds the current chromapod being fired while the PrintingDots status bit is set.
PrintingNozzles	4	This holds the current nozzle being fired while the PrintingDots status bit is set.

The following pseudocode illustrates the logic required to load a printhead for a single line. Note that loading commences only after the LineSync pulse arrives. This is to ensure the data for the line has been prepared by the LLFU and is valid for the first transfer to the printhead.

```
Wait for LineSync
For TransfersRemaining = 800 to 0
   For I = 0 to SegmentGroups
      If (SetAllNozzles)
         Set all ColorData lines to be 1
      Else
         Place 36 bit input on 36 ColorData lines
      EndIf
      Pulse SRClock[I]
      Wait SegmentsPerXfer cycles
      Send ADVANCE signal
   EndFor
EndFor
```

22.3.5 Preheat and Cleaning Cycles

The Cleaning and Preheat cycles are simply accomplished by setting appropriate registers in the MJI:

- SetAllNozzles = 1
- Set the PulseDuration register to either a low duration (in the case of the preheat mode) or to an appropriate drop ejection duration for cleaning mode.
- Set NumLines to be the number of times the nozzles should be fired
- Set the Go bit and then wait for the Go bit to be cleared when the print cycles have completed.

The LSGU must also be programmed to send LineSync pulses at the correct frequency.

22.4 LINE LOADER/FORMAT UNIT

The line loader/format unit (LLFU) loads the dots for a given print line into local buffer storage and formats them into the order required for the Memjet printhead. It is responsible for supplying the pre-calculated nozzleEnable bits to the Memjet interface for the eventual printing of the page.

The printing uses a double buffering scheme for preparing and accessing the dot-bit information. While one line is being loaded into the first buffer, the pre-loaded line in the second buffer is being read in Memjet dot order. Once the entire line has been transferred from the second buffer to the printhead via the Memjet interface, the reading and writing processes swap buffers. The first buffer is now read and the second buffer is loaded up with the new line of data. This is repeated throughout the printing process, as can be seen in the conceptual overview of Figure 74.

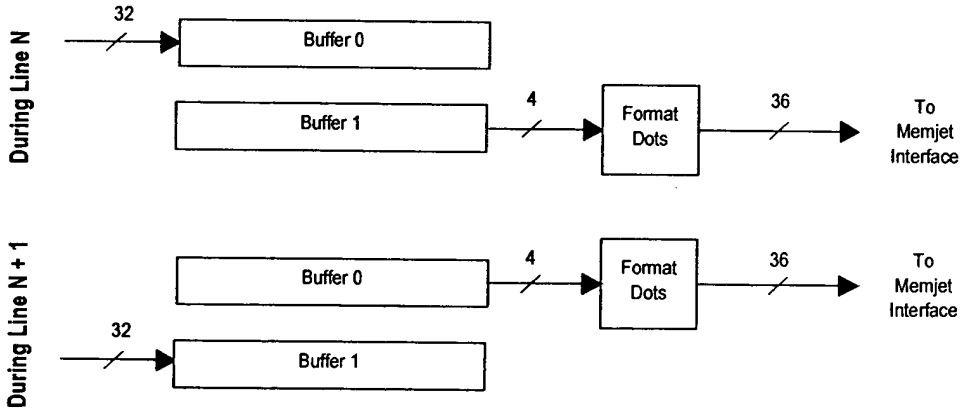


Figure 74. Conceptual Overview of Double Buffering During Print Lines N and N+1

The size of each buffer is 14KBytes to cater for the maximum line length of 18 inches in 4 colors (18 × 1600 × 4 bits = 115,200 bits = 14,400 bytes). The size for both Buffer 0 and Buffer 1 is 28.128 KBytes. While this design allows for a maximum print length of 18 inches, it is trivial to reduce the buffer size to target a specific application.

The actual implementation of the LLFU is shown in Figure 75. Since one buffer is being read from while the other is being written to, two sets of address lines must be used. The 32-bits DataIn from the common data bus are loaded depending on the WriteEnables, which are generated by the State Machine in response to the DMA Acknowledges.

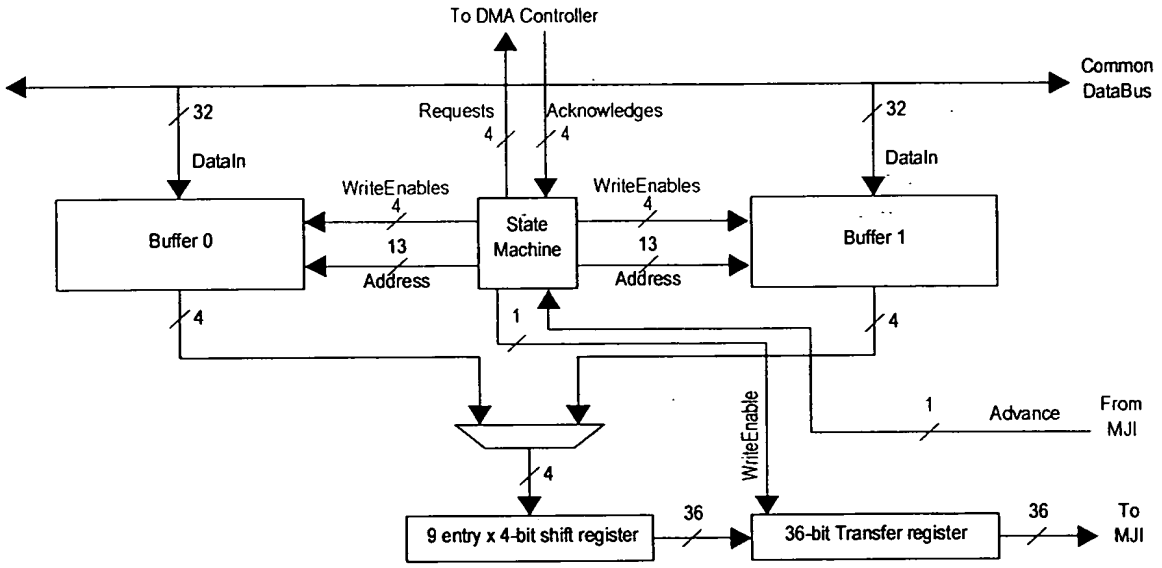


Figure 75. Structure of LLFU

A multiplexor chooses between the two 4-bit outputs of Buffer 0 and Buffer 1, and sends the result to a 9-entry by 4-bit shift register. After a maximum of 9 read cycles (the number depends on the number of segments written to per transfer), and whenever an Advance pulse comes from the MJI, the current 36-bit value from the shift register is gated into the 36-bit Transfer register, where it can be used by the MJI.

Note that not all the 36 bits are necessarily valid. The number of valid bits of 36 depends on the number of colors in the printhead, the number of segments, and the breakup of segment groups (if more than one segment group). For more information, see Section 21.2 on page 141.

A single line in an *L*-inch *C*-color printhead consists of ***1600L*** *C*-color dots. At 1 bit per colored dot, a single print-line consists of ***1600LC bits***. The LLFU is capable of addressing a maximum line size of 18 inches in 4 colors, which equates to 108,800 bits (14 KBytes) per line. These bits must be supplied to the MJI in the correct order for being sent on to the printhead. See Section 21.2.1 on page 142 for more information concerning the Load Cycle dot loading order, but in summary, 2LC bits are transferred to the printhead in ***SegmentGroups*** transfers, with a maximum of 36 bits per transfer. Each transfer to a particular segment of the printhead must load all colors simultaneously.

22.4.1 Buffers

Each of the two buffers is broken into 4 sub-buffers, 1 per color. The size of each sub-buffer is 3600 bytes, enough to hold 18-inches of single color dots at 1600 dpi. The memory is accessed 32-bits at a time, so there are 900 addresses for each buffer (requiring 10 bits of address).

All the even dots are placed before the odd dots in each color's buffer, as shown in Figure 76. If there is any unused space it is placed at the end of each color's buffer.

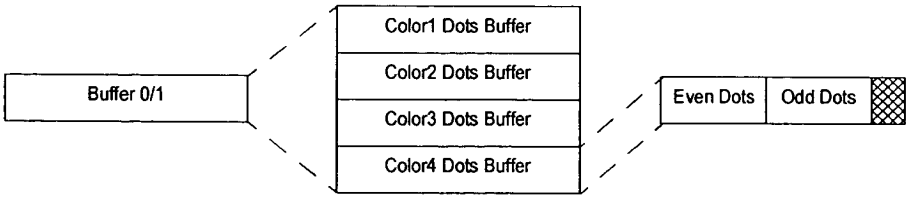


Figure 76. Conceptual Structure of Buffer

The *amount of memory* actually used is directly related to the printhead length. If the printhead is 18 inches, there are 1800 bytes of even dots followed by 1800 bytes of odd dots, with no unused space. If the printhead is 12 inches, there are 1200 bytes of even dots followed by 1200 odd dots, and 1200 bytes unused.

The *number of sub-buffers* gainfully used is directly related to the number of colors in the printhead. This number is typically 3 or 4, although it is quite feasible for this system to be used in a 1 or 2 color system (with some small memory wastage). In a desktop printing environment, the number of colors would be 4: Color1=Cyan, Color2=Magenta, Color3=Yellow, Color4=Black.

The addressing decoding circuitry is such that in a given cycle, a single 32-bit access can be made to all 4 sub-buffers - either a read from all 4 or a write to one of the 4. Only one bit of the 32-bits read from each color buffer is selected, for a total of 4 output bits. The process is shown in Figure 77. 15 bits of address allow the reading of a particular bit by means of 10-bits of address being used to select 32 bits, and 5-bits of address choose 1-bit from those 32. Since all color buffers share this logic, a single 15-bit address gives a total of 4 bits out, one per color. Each buffer has its own WriteEnable line, to allow a single

32-bit value to be written to a particular color buffer in a given cycle. The 32-bits of DataIn are shared, since only one buffer will actually clock the data in.

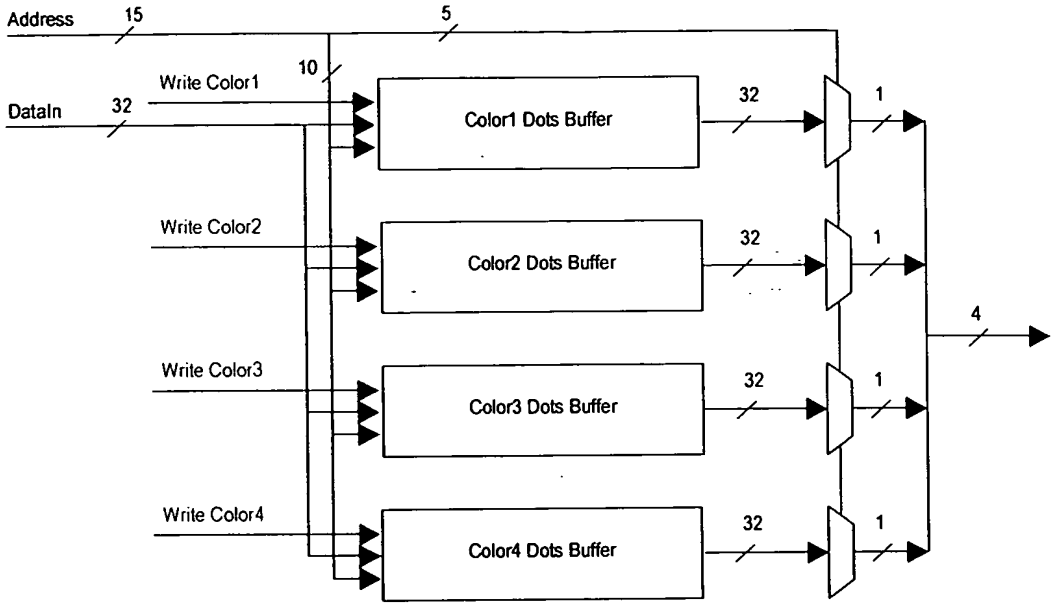


Figure 77. Logical Structure of Buffer

Note that regardless of the number of colors in the printhead, 4 bits are produced in a given read cycle (one bit from each color's buffer).

22.4.2 Address Generation

22.4.2.1 Reading

Address Generation for reading is straightforward. Each cycle we generate a bit address which is used to fetch 4 bits representing 1-bit per color for a particular segment. By adding 400 to the current bit address, we advance to the next segment's equivalent dot. We add 400 (not 800) since the odd and even dots are separated in the buffer. We do this firstly SegmentGroups sets of SegmentsPerXfer times to retrieve the data representing the even dots (the dot data is transferred to the MJI 36 bits at a time) and another SegmentGroups sets of SegmentsPerXfer times to load the odd dots. This entire process is repeated 400 times, incrementing the start address each time. Thus all dot values are transferred in the order required by the printhead in 400 × 2 × SegmentGroups × SegmentsPerXfer cycles.

In addition, we generate the TransferWriteEnable control signal. Since the LLFU starts before the MJI, we must transfer the first value before the Advance pulse from the MJI. We must also generate the next value in readiness for the first Advance pulse. The solution is to transfer the first value to the Transfer register after SegmentsPerXfer cycles, and then to stall SegmentsPerXfer-cycles later, waiting for the Advance pulse to start the next SegmentsPerXfer cycle group. Once the first Advance pulse arrives, the LLFU is synchronized to the MJI. However, the LineSync pulse to start the next line must arrive at the MJI at least 2SegmentsPerXfer cycles after the LLFU so that the initial Transfer value is valid and the next 32-bit value is ready to be loaded into the Transfer register.

The read process is shown in the following pseudocode:

```
DoneFirst = FALSE
For DotInSegment0 = 0 to 400
   CurrAdr = DotInSegment0
   XfersRemaining = 2 x SegmentGroups
   DotCount = SegmentsPerXfer
   Do
      V1 = DotCount = 0
      TransferWriteEnable = (V1 AND NOT DoneFirst) OR ADVANCE
      Stall = V1 AND (NOT TransferWriteEnable)
      If (NOT Stall)
         Shift Register=Fetch 4-bits from CurrReadBuffer:CurrAdr
         CurrAdr = CurrAdr + 400
         If (V1)
            DotCount = SegmentsPerXfer - 1
            XfersRemaining = XfersRemaining - 1
         Else
            DotCount = DotCount - 1
         EndIf
      EndIf
   Until (XfersRemaining=0) AND (NOT Stall)
EndFor
```

The final transfer may not be fully utilized. This occurs when the number of segments per transfer does not divide evenly into the actual number of segments in the printhead. An example of this is the 8½" printhead, which has 17 segments. Transferring 9 segments each time means that only 8 of the last 9 segments will be valid. Nonetheless, the timing requires the entire 9th segment value to be generated (even though it is not used). The actual address is therefore a don't care state since the data is not used.

Once the line has finished, the CurrReadBuffer value must be toggled by the processor.

22.4.2.2 Writing

The write process is also straightforward. 4 DMA request lines are output to the DMA controller. As requests are satisfied by the return DMA Acknowledge lines, the appropriate 8-bit destination address is selected (the lower 5 bits of the 15-bit output address are *don't care* values) and the acknowledge signal is passed to the correct buffer's WriteEnable control line (the Current Write Buffer is ¬CurrentReadBuffer). The 10-bit destination address is selected from the 4 current addresses, one address per color. As DMA requests are satisfied the appropriate destination address is incremented, and the corresponding TransfersRemaining counter is decremented. The DMA request line is only set when the number of transfers remaining for that color is non-zero.

The following pseudocode illustrates the Write process:

```
CurrentAdr[1-4] = 0
While (ColorXfersRemaining[1-4] are non-zero)
   DMARequest[1-4] = ColorXfersRemaining[1-4] NOT = 0
   If DMAAknowledge[N]
      CurrWriteBuffer:CurrentAdr[N] = Fetch 32-bits from data bus
      CurrentAdr[N] = CurrentAdr[N] + 1
      ColorXfersRemaining[N] = ColorXfersRemaining[N] - 1 (floor 0)
   EndIf
EndWhile
```

22.4.3 Registers

The following interface registers are contained in the LLFU:

Table 30. Line Load/Format Unit Registers

Register Name	Description
SegmentsPerXfer	The number of segments whose dots must be loaded before each transfer. This has a maximum value of 9.
SegmentGroups	The number of segment groups in the printhead. This has a maximum number of 4.
CurrentReadBuffer	The current buffer being read from. When Buffer0 is being read from, Buffer1 is written to and vice versa. Should be toggled with each AdvanceLine pulse from the MJI.
Go	Bits 0 and 1 control the starting of the read and write processes respectively. A non-zero write to the appropriate bit starts the process.
Stop	Bits 0 and 1 control the stopping of the read and write processes respectively. A non-zero write to the appropriate bit stops the process.
Stall	This read-only status bit comes from the LLFU's Stall flag. The Stall bit is valid when the write Go bit is set. A Stall value of 1 means that the LLFU is waiting for the ADVANCE pulse from the MJI to continue. The CPU can safely start the LSGU for the first line once the Stall bit is set.
ColorXfersRemaining[1-4]	The number of 32-bit transfers remaining to be read into the specific Color[N] buffer.

22.5 CONTROLLING A PRINT

When controlling a print the CPU programs and starts the LLFU in read mode to ensure that the first line of the page is transferred to the buffer. When the interrupts arrive from the DMA controller, the CPU can switch LLFU buffers, and program the MJI. The CPU then starts the LLFU in read/write mode and starts the MJI. The CPU should then wait a sufficient period of time to ensure that other connected printer controllers have also started their LLFUs and MJIs (if there are no other connected printer controllers, the CPU must wait until the Stall bit of the LLFU is set, a duration of 2SegmentsPerXfer cycles). The CPU can then program the LGSU to start the synchronized print. As interrupts arrive from the DMA controllers, the CPU can reprogram the DMA channels, swap LLFU buffers, and restart the LLFU in read/write mode. Once the LLFU has effectively filled its pipeline, it will stall until the next Advance pulse from the MJI. The MJI does not have to be touched during the print.

If for some reason the CPU wants to make any changes to the MJI or LLFU registers during an inter-line period it should ensure that the current line has finished printing/loading by polling the status bits of the MJI and the Go bits of the LLFU.

REFERENCES

23. Silverbrook References

1. Patent Cooperation Treaty Patent WO 99/04368

2. Patent Cooperation Treaty Patent WO 99/04368

3. Australian Provisional Patent Number PP7738 entitled "An Image Creation Method and Apparatus (CEP01)"

4. Australian Provisional Patent Number PP9223 entitled "A Method and Apparatus (IJ46P2)"

24 Other References

[5] Adobe Systems Inc., *Portable Document Format Reference Manual*, version 1.2, November 27, 1996

[6] Adobe Systems Inc., *PostScript Language Reference Manual*, Second Edition, 1990

[7] ANSI/EIA 538-1988, *Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Equipment*, August 1988

[8] Bank of Montreal Economics Department, "Newsprint Market Update", Forest Products, May 1998

[9] Bender, W., "Read all about it in the Daily You", Communicating Business, Forward Publishing, London, Winter 1994/95 (http://nif.www.media.mit.edu/papers/forward.html)

[10] Bender, W., P. Chesnais, S. Elo, A. Shaw, and M. Shaw, "Enriching communities: Harbingers of news in the future", IBM Systems Journal, Vol.35, Nos.3&4, 1996, pp.369-380

[11] Brand, S., "Personal Newspaper", in *The Media Lab*, Penguin Books, 1988, pp.36-39

[12] Bray, T., "News Wire Services Heading for XML", 12 August 1998, http://xml.com/xml/pub/98/08/nitf.html

[13] Cable Datacom News, "Cable Modem Customer Count To Top 500,000 At Year's End", December 1998, http://www.cabledatacomnews.com/dec98-1.htm

[14] Cable Foreman Xtra, "WDM boosts fiber-optic capacity", May 1998, http://www.broadband-guide.com/cbl4man/eng/eng5981.html

[15] Chesnais, P.R., M.J. Mucklo and J.A. Sheena, "The Fishwrap Personalized News System", *IEEE Second International Workshop on Community Networking*, Integrating Multimedia Services to the Home, 1995

[16] Cisco Systems, "Cisco 605 Personal PCI ADSL Modem", http://www.cisco.com/warp/public/728/605

[17] Cisco Systems, "Cisco 6100 Series advanced DSL Access Multiplexer", http://www.cisco.com/warp/public/728/6100

[18] Clark, T., "Visa, Mastercard try to revive SET", *CNET NEWS.COM*, 12 May 1999, http://www.news.com/News/Item/0,4,0-36435,00.html@st.ne.180.head

[19] Click~Through, http://www.clickthrough.com

[20] CNN, *CNN Custom News*, http://customnews.cnn.com/cnews/pna_auth.welcome

[21] CNN Interactive, "With 'electronic ink', a printed page can be a work in progress", http://cnn.com/TECH/science/9812/23/electronic.ink.ap/index.html

[22] Cook, P.S., D. Gomery, and L.W. Lichty (eds.), *The Future of News - Television, Newspapers, Wire Services, Newsmagazines*, The Woodrow Wilson Center Press, 1992

[23] Crothers, B., "IBM eyes new way to sell PCs", *CNET NEWS.COM*, 14 May 1999, http://www.news.com/News/Item/0,4,0-36508,00.html@st.ne.ni.lh

[24] Daniel, R., and Mealling, M., "Resolution of Uniform Resource Identifiers using the Domain Name System", *Internet Engineering Task Force RFC2168*, June 1997

[25] Doane, W.J., "A Comparison of Display Technologies for E-Books", http://www.jmc.kent.edu/futureprint/1998fall/doane.htm

[26] E Ink, http://www.eink.com

[27] Ericsson, *3G Cellular Technology*, http://www.ericsson.se/wireless/products/modsys/3rdgen/subpages/3gcell

[28] E&P Interactive, "Online News Reading Soars", *E&P Interactive*, 12 June 1998, http://www.mediainfo.com/ephome/news/newshtm/stories/061298n4.htm

[29] E&P Interactive, "Audience Rather than Circulation", *E&P Interactive*, 24 July 1998, http://www.mediainfo.com/ephome/news/newshtm/stories/072498n4.htm

[30] E&P Interactive, "Newspapers Near Death, Gates Says", *E&P Interactive*, 5 February 1999, http://www.mediainfo.com/ephome/news/newshtm/stories/020599n1.htm

[31] Farrell, J., "How to Allocate Bits to Optimize Photographic Image Quality", Proceedings of IS&T International Conference on Digital Printing Technologies, 1998, pp.572-576

[32] Fishkin, K., "Filling a Region in a Frame Buffer", *Graphics Gems*, Academic Press 1990, pp.278-284

[33] Fitzgerald, M., "Newspapers Re-think Microzoning", *E&P Interactive*, 14 January 1999, http://www.mediainfo.com/ephome/news/newshtm/stories/011499n3.htm

[34] Gannet Co. Inc., *Consolidated Statements of Income 1998*, http://www.gannet.com/annual/ar98/financials/income.htm
[35] Handley, K., "All the World's a Page", http://www.ingersoll-rand.com/compair/oct_nov_97/octnovco/pag_1.htm
[36] Heckbert, P.S., "A Seed Fill Algorithm", *Graphics Gems*, Academic Press 1990, pp.275-277
[37] Heinrich, J., *MIPS R4000 Microprocessor User's Manual*, PTR Prentice Hall, 1993
[38] Hoffert, E.M. and G. Gretsch, "The Digital News System at EDUCOM", Communications of the ACM, Vol.34, No.4, April 1991, pp.113-116
[39] Hughes Network Systems, "Hughes Spaceway - Wireless Broadband on Demand", http://www.hns.com/spaceway/spaceway.htm
[40] Humphreys, G.W., and V. Bruce, *Visual Cognition*, Lawrence Erlbaum Associates, 1989, p.15
[41] INRIA, "Light-weight Reliable Multicast Protocol", http://monet.inria.fr/lrmp/lrmp.html
[42] INRIA, "WebCanal White Paper - Global Information Broadcast", http://monet.inria.fr/white/index.html
[43] IP Multicast Initiative (IPMI), *Higher Level Protocols used with IP Multicast*, 1997, http://www.ipmulticast.com/...
[44] IPTC - NAA, *News Industry Text Format*, version 2.0b1, June 1998
[45] ISO/IEC 19018-1:1994, *Information technology - Digital compression and coding of continuous-tone still images: Requirements and guidelines*, 1994
[46] ISO/IEC JTC1/SC29/WG1 (Joint Photographic Experts Group), *Call for contributions for JPEG 2000: Image Coding System*, http://www.jpeg.org/public/wg1n505.pdf
[47] ISO/IEC JTC1/SC29/WG1 (Joint Photographic Experts Group), *JPEG 2000 requirements and profiles version 4.0*, http://www.jpeg.org/public/wg1n1105.pdf
[48] Kent Displays, http://www.kentdisplays.com
[49] Krause, J.K., "Two More 'Free' PC Firms Pin Hopes on E-commerce", *The Industry Standard*, 1 April 1999, http://wwwthestandard.net/articles/article_print/0,1454,4073,00.html
[50] Lambert, P., "Multicast Debuts on Cable Modems", *ZDNN*, 5 August 1997, http://www5.zdnet.com/zdnn/content/inwk/0426/inwk0006.html
[51] Liebeskind, K., "Newspapers Continue to Report Higher Revenues", E&P Interactive, 8 January 1998, http://www.mediainfo.com/ephome/news/newshtm/stories/010899n1.htm
[52] Loberg, C.J., "Testing the Future of All-Optical Networks", Lightwave Xtra, March 1998, http://www.broadband-guide.com/lw/reports/report3981.html
[53] Loeffler, C., A. Ligtenberg and G. Moschytz, "Practical Fast 1-D DCT Algorithms with 11 Multiplications", Proceedings of the International Conference on Acoustics, Speech, and Signal Processing 1989 (ICASSP '89), pp.988-991
[54] Lyppens, H., "Reed-Solomon Error Correction", *Dr. Dobb's Journal* Vol.22, No.1, January 1997
[55] McCanne, S., "Overview of Reliable Multicast Protocols", *Matters Mbone Workshop*, SIGCOMM '96, August 1996, http://gaia.cs.umass.edu/sigcomm_mcast/talk1.html
[56] McCanne, S., "SRM: Work in Progress", *Matters Mbone Workshop*, SIGCOMM '96, August 1996, http://gaia.cs.umass.edu/sigcomm_mcast/talk2.html
[57] Marimba, *Castanet 3.2 Family Brochure*, http://www.marimba.com/products/resource.htm
[58] MasterCard/Visa, *SET Secure Electronic Transaction Specification, Book 1: Business Description*, Version 1.0, 31 May 1997, http://www.setco.org/set_specifications.html
[59] Medin, M., "Data Over Cable and the @Home Network", http://www.academ.com/nanog/feb1996/data.over.cable.html
[60] MIT Media Lab, *News in the future*, http://nif.www.media.mit.edu/nif.html
[61] MIT Media Lab, *Fishwrap Documentation Center*, http://fishwrap-docs.www.media.mit.edu/docs
[62] MIT Media Lab, *Micromedia - Electronic Paper* http://www.media.mit.edu/micromedia/elecpaper.html
[63] My Yahoo, http://my.yahoo.com
[64] Negroponte, N., "Less is More", in *Being digital*, Vintage Books, 1996, pp.149-159
[65] New Media Communication, *CyberStream Fast Internet & Data Broadcasting System via Satellite*, http://www.nmc.com

[66] New York Times Co., *Consolidated Statements of Income 1998*, http://www.nytco.com/pdf/consol1_1998.pdf
[67] Neuwirth, R., "Inserts Outpace ROP", *E&P Interactive*, 24 September 1998, http://www.mediainfo.com/ephome/news/newshtm/stories/092598n1.htm
[68] Newspaper Association of America (NAA), *Facts About Newspapers 1998* http://www.naa.org/info/facts
[69] Noack, D., "Online News Audience Becomes More Mainstream", *E&P Interactive*, 15 January 1999, http://www.mediainfo.com/ephome/news/newshtm/stories/011599n2.htm
[70] Platt, C., "Digital Ink", *Wired* magazine, May 1997
[71] Rorabaugh, C, *Error Coding Cookbook*, McGraw-Hill 1996
[72] Schneier, S., "The Crypto Bomb is Ticking", *Byte Magazine*, May 1998, pp.97-102
[73] Schwartz, E.I., "Advertising Webonomics 101", *Wired* magazine, February 1996
[74] Shardanand, U. and Maes, P., "Social Information Filtering: Algorithms for Automating 'Word of Mouth'", *Proceedings of ACM Computer-Human Interaction (CHI) 1995*
[75] Steinberg, D., "Broading the Bandwidth to the Home in USA and Canada", 15 May 1997, http://www.specialty.com/hiband/mi-int.html
[76] Sterne, J., *Advertising on the Web*, Que 1997
[77] Stevens, W.R., *TCP/IP Illustrated, Volume 1 - The Protocols*, Addison-Wesley 1994
[78] Stone, M., "Newspapers Must Change, Fast - API Cites Serious Threat of Online Classifieds", *E&P Interactive*, 24 July 1998, http://www.mediainfo.com/ephome/news/newshtm/stories/072498n1.htm
[79] Sun Microsystems, http://www.sun.com
[80] Tappert, C.C., Suen, C.Y., Wakahara, T., "The State of the Art in On-Line Handwriting Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.12, No.8, August 1990
[81] Teledesic, *Teledesic Overview*, http://www.teledesic.com/overview.html
[82] The New York Times on the Web, http://www.nytimes.com
[83] The Unicode Consortium, *The Unicode Standard, Version 2.0*, Addison Wesley 1996
[84] Thompson, H.S., *Multilingual Corpus 1* CD-ROM, European Corpus Initiative
[85] Turkowski, K., "Filters for Common Resampling Tasks", *Graphics Gems*, Academic Press 1990, pp.147-165
[86] United States Postal Service, *Postal Facts May 1998*, http://www.usps.gov/history/pfact98.htm
[87] Urban, S.J., "Review of standards for electronic imaging for facsimile systems", Journal of Electronic Imaging, Vol.1(1), January 1992, pp.5-21
[88] Wallace, G.K., "The JPEG Still Picture Compression Standard", Communications of the ACM, Vol.34, No.4, April 1991, pp.30-44
[89] Weinberger, D., "The Daily Me? No, the Daily Us", *Wired* magazine, April 1995
[90] Welch Allyn Inc., *Aztec Barcode Symbology Specification Rev 3.0*, 1995 http://dcd.welchallyn.com/techover/dcdwhite.htm
[91] Wicker, S., and Bhargava, V., *Reed-Solomon Codes and their Applications*, IEEE Press 1994
[92] World Wide Web Consortium, *Extensible Markup Language (XML) 1.0*, W3C Recommendation, 10 February 1998, http://www.w3.org/TR/1998/REC-xml-19980210
[93] World Wide Web Consortium, *Extensible Stylesheet Language (XSL) Specification*, W3C Working Draft, 21 April 1999, http://www.w3.org/TR/WD-XS/
[94] World Wide Web Consortium, *HTML 4.0 Specification*, W3C Recommendation, 24 April 1998, http://www.w3.org/TR/REL_html40/
[95] World Wide Web Consortium, *Scalable Vector Graphics (SVG) Specification*, W3C Working Draft, 12 April 1999, http://www.w3.org/TR/WD-SVG/
[96] Yahoo, http://www.yahoo.com
[97] Yasuda, Y., "Overview of Digital Facsimile Coding Techniques in Japan", Proceedings of the IEEE, Vol. 68(7), July 1980, pp.830-845
[98] Zollman, P.M., "1999 Could make or Break News Web Sites", *E&P Interactive*, 16 December 1998, http://www.mediainfo.com/ephome/news/newshtm/stories/121698n1.htm